App Orchard Web Services

Last Reviewed: September 19, 2017
# Table of Contents

- Overview.................................................................................................................... 3
- **Web Services**............................................................................................................ 3
  - Access.......................................................................................................................... 3
  - Account......................................................................................................................... 5
  - Billing............................................................................................................................ 5
  - Clinical.......................................................................................................................... 6
  - Common......................................................................................................................... 11
  - Core............................................................................................................................... 12
  - CRM............................................................................................................................... 13
  - EMPI............................................................................................................................... 14
  - FHIR............................................................................................................................... 14
  - Imaging........................................................................................................................... 16
  - MyChart........................................................................................................................ 16
  - Orders............................................................................................................................. 19
  - Patient Access............................................................................................................... 19
  - Patient Access – Mobile.............................................................................................. 21
  - Pharmacy...................................................................................................................... 25
  - Scheduling..................................................................................................................... 25
  - Security......................................................................................................................... 27
  - Specialty....................................................................................................................... 29
## Overview

Epic provides dozens of fully encapsulated business services that you can use to interact with Epic data and meet specific business requirements for your applications. Get technical specifications for these services through our [App Orchard developer program](http).

## Web Services

### Access

**AssignVisitAccountAndCoverage**
Assigns the provided guarantor account and coverage as the visit account and coverage for a provided contact and patient, and returns the copay due.

**CompleteRequest**
Marks an environmental service (EVS) cleaning request as complete.

**CreateDisclosure**
Imports disclosure data and creates a Third Party Disclosure record in Epic.

**CreateRelease**
Creates a new release record and returns the record ID.

**EndShift**
Ends the shift and logs out a housekeeper.

**GetCensusByUnit**
Returns the census information for a specified unit.

**GetEVSCategoryInfo**
Returns the hold reason and delay reason, both of which are required for Environmental Services (EVS) cleaning requests.

**GetGenericRegNoAddCategory**
Returns custom registration information when given the patient ID, ID type, and the Epic database items to retrieve.

**GetGuarantorsAndCoverages**
Returns the patient’s guarantor accounts and coverages available for selection on a visit for the current day.

**GetGuarantorsAndCoverages2**
Returns the patient’s guarantor accounts and coverages available for selection on a visit for the current day, including relevant information about those guarantors and coverages. Guarantors available for selection
include those with at least one current or future effective coverage, as well as inactive guarantors with at least one current or future effective coverage.

*GetHousekeeperInfo*

Returns the current Environmental Services (EVS) cleaning requests for a housekeeper.

*GetHousekeeperLoginInfo*

Returns the login information for a housekeeper, so a programmer knows the records and categories that can be used with the StartShift web service.

*GetPatientLocation*

Returns the in-house location and admission information for admitted patients.

*GetPatientLocationByVisit*

Returns basic patient and visit information needed to produce a label.

*GetPatientReferrals*

Returns a patient’s referrals. Options for filtering include referral status, service area, providers associated with the referral, and date range.

*GetPersonInfo*

Enables external systems that don’t receive ADT information from Epic to query Epic for the following patient information:

- Demographics
- Emergency contacts
- PCP
- Open and recent hospital accounts (Danish kontakt)

*GoOffBreak*

Marks a housekeeper as off break in Grand Central’s EVS Module.

*GoOnBreak*

Marks a housekeeper as on break in Grand Central’s EVS Module.

*MarkInProgress*

Marks an environmental service cleaning request as in progress.

*PutRequestOnDelay*

Marks environmental cleaning requests as delayed.

*PutRequestOnHold*

Marks environmental cleaning requests as on hold.
**StartShift**
Starts a shift for a housekeeper in Grand Central’s EVS Module.

**UpdateGenericRegNoAddCategory**
Updates custom registration information when given the patient ID, ID type, and the Epic database items to update.

**UpdatePatientDemographics**
Updates a patient’s demographic information when given a patient ID, patient ID type, user ID, and user ID type.

**UpdatePatientVerification**
Runs patient verification for a given patient and encounter, and returns any errors and warnings that occur.

**UpdateReferral**
Updates an existing referral.

**UpdateRelease**
Updates an existing release record in Epic.

**Account**

**GetAccountDemographics**
Returns the guarantor account’s demographics information.

**GetAccountPayments**
Returns an account’s recent payments.

**GetPatientBillingDetails**
Returns a patient’s billing details.

**Billing**

**CreateChargeSession**
Creates a group of Charge Router charge records, saves the charges in a new charge session, and submits the charges to the Charge Router.

**GetBillingPatientFromGuarantor**
Intended for use with an IVR system. Returns the associated billing patient ID and patient ID type for a given guarantor ID.

**GetBillingPatientFromHospitalAccount**
Intended for use with an IVR system. Returns the associated billing patient ID and patient ID type for a given hospital account ID.
GetCoverageBuckets

Retrieves coverage, plan, and benefit bucket limit and accumulation information for a subset of coverages linked to a specific patient. If you need to request benefit bucket information for multiple coverages with one call, use the GetCoverageBucketsMultiple web service, which uses this web service to return the coverage information.

GetEncounterCharges

Returns the charges from a patient's encounter.

GetPatientFlags

Returns all patient flags (patient FYIs) associated with the patient.

PhoneSystem

Sends a request to your phone system to pause the recording when the Hyperspace credit card entry form opens, and sends another request to the phone system to resume the recording when the Hyperspace credit card form closes.

SendAccountLetter

Intended for use with an IVR system. Returns a SmartText letter for the specified account level for a given hospital or guarantor account ID. The default is a hospital-level letter.

SendHospitalAccountDetailBill

Intended for use with an IVR system. Returns a detail bill for a given hospital account ID.

SetPatientFlag

Sets the patient flag.

VoidCharge

Voids a Charge Router charge record.

Clinical

AddAllergy

Adds an allergy to a patient’s chart.

AddDiagnoses

Adds visit diagnosis information. For each diagnosis, you can specify the diagnosis ID, description or display name, comments, qualifier, and chronicity. You can also specify whether the first diagnosis filed is the primary diagnosis.

AddDiagnoseToMedicalHistory

Adds a new diagnosis record to the patient’s medical history. You can add important details about the diagnosis to the medical record, such as an annotation, comment, and diagnosis date.
**AddDxAssociation**

Adds a diagnosis association to an order.

**AddFlowsheetValue**

Adds a flowsheet value and comment to the row on the template for the appropriate patient contact.

**AddHealthMaintenanceModifier**

Adds a Health Maintenance modifier to a patient.

**AddLOS**

Adds a Level of Service (LOS) code and related information for an outpatient visit. Additional EM codes can also be added. The LOS and EM codes can be associated with existing diagnoses as well as new diagnoses filed to the visit.

**AddMedicationRefills**

Adds refills to an ambulatory medication order without reordering the medication order. Note that this web service requires either the number of refills or a PRN indicator. If a PRN indicator is passed, the end date is required must be a future date (relative to the service request). If a PRN indicator is passed, a PRN valid for one year is assumed.

**AddPatientsToMyPatientList**

Adds patients to an existing patient list if the patient isn’t already on a patient list.

**AddProblems**

Adds the specified diagnoses to the patient’s problem list.

**AddProcedureToSurgicalHistory**

Adds a new procedure to the patient’s surgical history. You can specify important details about the procedure, such as the date, ID, comment, and laterality.

**AddToProblemList**

Adds one or more problems to a patient’s problem list. Can also file hospital problems.

**CareGap**

Stores a patient’s Care Gaps in Epic released SDEs.

**CreateMyPatientList**

Creates a user-specific patient list and add patients to the list.

**CreatePatientEvent**

Creates an event for a patient, which can be used to support notification buttons in an ICU setting.

**EditProblems**

Edits an existing problem on a patient’s problem list.
**EditProviderProcedureAuthorizations**

Edits a surgeon’s authorizations to perform certain procedures in certain locations. This web service specifically handles editing of the pre-2015 OpTime authorizations structure.

**EditProviderProcedureKeyAuthorizations**

Edits a surgeon’s authorizations to perform certain procedures in certain locations. This web service specifically handles editing of the 2015+ OpTime authorizations structure.

**GenerateLettersFromTemplate**

Generates patient letters from patient panels in an external system. When letters are generated from the external system, generic letters appear in Epic, so users viewing a patient’s chart are aware that a letter has been sent. The reason for letter and any comments also appear.

**GetActiveAllergies**

Returns the active allergies for the patient.

**GetActiveMedications**

Returns the active medications for the patient.

**GetActiveProblemList**

Returns the list of active problems for the patient.

**GetActiveProblems**

Returns the active problems for a patient.

**GetComponentMetadata**

Returns data stored in a component record.

**GetCurrentMedications**

Returns the patient’s current outpatient medications, current inpatient medications, or both.

**GetCurrentMedicationsByGrouper**

Returns a list of the patient’s current medications, filtered by grouper ID.

**GetDeviceData**

Returns device data for a patient contact.

**GetFamilyHistory**

Returns a patient’s family history.

**GetFlowsheetRows**

Returns flowsheet rows from a patient’s chart.
GetHealthMaintenanceTopics
Returns a list of Health Maintenance topics for the patient.

GetIntakeOutputTotals
Returns intake/output totals from a patient's chart.

GetMedicalHistory
Returns a patient's medical history.

GetMedicationAdministrationHistory
Returns a patient’s medication administration history.

GetMyPatientLists
Returns all user-specific patient lists for a single user. The name and identifier for the list is returned as the response to this request.

GetPatientContactInformation
Returns information about an existing contact ID. Contacts are limited to contacts available to the user in Chart Review and Media Manager.

GetPatientContacts
Returns a list of patient contacts. Contacts are limited to contacts available to the user in Chart Review and Media Manager.

GetPatientResultComponents
Returns an array of results for a single component record for a single patient.

GetPatientsOnMyList
Returns the patients on a My List and the System Lists contained in that My List. For each patient on the list, the service returns the patient IDs, name, age, and sex. If the requested My List contains a System List with unoccupied beds, the unoccupied bed rows are not returned.

GetPatientsOnSystemList
Returns the patients on a System List. For each patient on the list, the service returns the patient IDs, name, age, and sex. If the requested System List has unoccupied beds, the unoccupied bed rows are not returned.

GetPreferredAndRecentPharmacies
Returns a list of the patient's preferred and recently used pharmacies.

GetResultsForOrder
Returns result information for a single order.

GetResultsForProcedure
Returns result information for a particular procedure.
**GetSmartDataValues**
Returns raw values for SmartData elements.

**GetSurgicalHistory**
Returns a patient’s surgical history.

**GetTreatmentTeam**
Returns a patient’s treatment team, including attending and admitting providers.

**OverrideHealthMaintenanceTopic**
Overrides a Health Maintenance topic for a patient.

**PostponeHealthMaintenanceTopic**
Postpones a Health Maintenance topic for a patient.

**RemoveDiagnoseFromMedicalHistory**
Removes a diagnosis from a patient’s medical history.

**RemoveHealthMaintenanceModifier**
Removes a Health Maintenance modifier for a patient.

**RemovePatientsFromMyPatientList**
Removes patients from an existing patient list.

**RemoveProcedureFromSurgicalHistory**
Removes a procedure from a patient’s surgical history.

**SearchPharmacies**
Searches for pharmacies. For example, you can find pharmacies located near the patient, as determined by the patient’s zip code.

**SetPreferredAndEncounterPharmacy**
Adds a pharmacy to the patient’s list of preferred pharmacies and optionally sets the pharmacy being used in an encounter.

**SetSmartDataValues**
Sets raw values for SmartData elements.

**UpdatePatientCareTeam**
Updates a patient’s care team.

**UpdateDiagnosesInMedicalHistory**
Updates an existing diagnosis in the patient’s medical record. For example, you can update the annotation, comment, date, or ID of a diagnosis.
**UpdateProcedureInSurgicalHistory**

Updates an existing procedure in the patient’s surgical history. For example, you can update the comment, date, ID, or laterality of a procedure.

**ViewProviderProcedureAuthorizations**

Returns a surgeon’s authorizations to perform certain procedures in certain locations. This web service specifically handles viewing of the pre-2015 OpTime authorizations structure.

**ViewProviderProcedureKeyAuthorizations**

Returns a surgeon’s authorizations to perform certain procedures in certain locations. This web service specifically handles viewing of the 2015+ OpTime authorizations structure.

**Common**

**AddDevice**

Adds an authorized handheld device for Haiku or Rover.

**BatchResultsAckSubmitTransaction**

Accepts incoming acknowledgements for batch ANSI 5010 X12 results.

**BatchResultsRetrievalTransaction**

Responds with the results of batch ANSI 5010 X12 submissions.

**DeleteDevice**

Deletes a previously authorized device for Haiku.

**GetInternalIdentifier**

Receives the ID, ID type, and master file for the ID and returns and internal identifier.

**GetPatientDemographics**

Returns a patient’s demographics information.

**GetPatientDemographics (Epic 2015 version)**

Returns the patient’s demographics.

**GetPatientChartUpToDate**

Queries whether the patient’s chart is up to date. If the chart is not up to date and synchronization has not started, the web service starts the process of synchronizing the patient chart and returns a time estimate in seconds of how long it expects the synchronization will take. If the chart is not up to date but synchronization has started, the web service returns the status and time estimate but doesn’t restart synchronization.

**GetPatientIdentifiers (Epic 2010 version)**

Returns all patient identifiers.
**GetPatientIdentifiers (Epic 2014 version)**

Returns the list of active IDs as well as the list of historical IDs for a patient.

**GetPatientIdentifiers (Epic 2015 version)**

Returns the list of active IDs and historical IDs for a patient.

**GetPatientPhoto**

Returns the patients profile photo.

**RealTimeTransaction**

Responds to incoming real-time ANSI 5010 X12 inquiries.

**ReceiveCommunication**

When a user receives a phone call, the phone system alerts the database with the information it has, so that a contact can be logged and the appropriate activity launched. If a contact type is specified, a new communication tracking (CAL) record is created to document the phone call itself, which is attached to the record of the launched activity. If call information is not provided, this web service can still be used to launch any patient related activity. There will be no response, only error codes. On success, there is nothing that needs to be returned.

**SetLoggedErrorDeleted**

Soft-deletes logged errors.

**SetLoggedErrorUndeleted**

Restores soft-deleted logged errors.

**SetStatus**

Activates or deactivates an authorized handheld device for Haiku or Rover.

**UpdateUserDemographics**

Updates user records. This service can also optionally update the same demographic information in the linked provider record.

**VerifyPatient**

Returns a boolean value that indicates whether information in the message matches information on patient’s record.

**Core**

**AddPrinter**

Creates a printer device with specified configurations.
**CheckPrinterRedirection**
Validates and resolves the printer redirection chain. This web service can be used to look up an existing printer redirection chain, or validate a redirection chain after the user attempts to make a change.

**DeletePrinter**
Soft-deletes a printer device.

**EditPrinter**
Edits the configuration of a printer device.

**GetDeviceTypeList**
Returns a list of all the available device types.

**GetImportDataLog**
Reads and returns the contents of the Chronicles Import Log.

**GetPrinterDetails**
Returns printer device details.

**GetPrinterList**
Returns printers in the database. Can also be used for searching.

**ImportData**
Uses Chronicles import specifications to file data into Epic.

**PrintTestPage**
Prints test pages from a printer device. Supports both rich text and plain text.

**RestorePrinter**
Restores a soft-deleted printer device.

**CRM**

**CreateCustomerServiceRecord**
Creates a CRM record.

**GetCustomerServiceStatus**
Returns the status of a CRM record.

**LogPatientCommunication**
Documents communications with patients.

**ResolvedCustomerServiceRecord**
Updates an existing CRM record with information relating to the resolution of a remote task.
**UpdateCustomerServiceRecord**

Updates a CRM record.

**EMPI**

**AssignPatientIdentifier**

Generates and assigns an MRN to a patient in Epic or assigns an MRN provided by the caller to a patient in Epic. If the MRN is successfully assigned, the patient’s ID table is returned.

**PatientCreate**

Creates patients and returns the patient’s demographic information as entered in the new record as well as Identity IDs that were created for the patient. You can have a custom or external application provide basic demographics such as name, sex, date of birth, and Social Security number (SSN) to this service, along with the user and login department. This service assumes that duplicate checking has already been done.

**PatientLookup (Summer 2009)**

Returns a list of patients with the demographics information that match the given search criteria. This service can be invoked from a Web application where the user needs to perform a patient search with certain criteria and then select that patient, get his home ID and deployment, and continue in the selected workflow.

**PatientLookup (Epic 2012)**

Returns a list of patients and the associated demographic information that match the given search criteria. This version of the service is designed to replace the Summer 2009 version of the PatientLookup service.

**PatientLookup2**

Looks up patients by matching on address, email, and phone numbers.

**FHIR**

**AllergyIntolerance (Read/Search)**

Returns data about an allergy or intolerance to a specific substance associated with one patient. The AllergyIntolerance resource can also accommodate search by ID and by patient, allowing it to return a list of allergies.

**Binary (Read Only)**

Looks up binary resources by a constant server ID. The read interaction allows clients to store only the server ID, and with a single request, retrieve the most up-to-date binary content. Read interactions typically begin with a client having previously established a relationship, often through querying for DocumentReference resources through the search interaction.

**CarePlan (Read/Search)**

The FHIR CarePlan resource is a broad container for summarizing the plan of treatment for a patient. It includes links to the Condition Resource (the patient’s long term Problem List) and the Goal Resource (the patient’s longitudinal goals) as well as in-line detail about upcoming appointments, referral orders, and upcoming orders. This resource is designed for stating specific activities related to a single patient, rather
than generic protocols for condition treatment (a Protocol resource will be defined in a later version of FHIR).

**Condition (Read/Search)**
Retrieves condition (diagnosis) data from a patient's chart. This includes any data found in the patient’s problem list across all encounters.

**Device (Read/Search)**
Returns information about durable, manufactured medical items. Currently, only implantable medical devices are supported.

**DiagnosticReport (Read/Search)**
Returns information about a diagnostic service performed on a patient. Currently, only laboratory reports are supported.

**DocumentReference (Read/Search)**
Returns information about a document.

**FamilyMemberHistory (Search Only)**
Returns the medical and surgical history of a patient's relatives.

**Goal (Read/Search)**
Defines objectives for a patient based on a current condition or recent procedure. The goal may be long term, such as targeting a specific HgbA1c level after a diabetes diagnosis, or shorter term such as changing wound dressings routinely following a procedure.

**Immunization (Read/Search)**
Returns a patient's immunizations, including vaccine and vaccine administration details.

**Medication (Read Only)**
Returns information about a specific medication. It serves as a matter of identifying an item from a list and then conveying a reference for the item selected either into a patient related resource or to other applications.

**MedicationOrder (Read/Search)**
Returns medication orders for both supply of the medication and the instructions for administration of the medicine.

**Observation (Read/Search)**
Returns information about a patient, device, or other subject including device measurements, vital signs, smoking status.

**Patient (Read/Search)**
Returns patient demographic information from a patient's chart. Patients can be looked up by a single encrypted ID representing the resource or by a search on a number of parameters.
Practitioner (Read Only)

Returns data about a practitioner given a FHIR ID for a direct read.

Procedure (Read Only)

The FHIR Procedure resource defines an activity performed on or with a patient as part of the provision of care. It corresponds with surgeries and procedures performed, including endoscopies and biopsies, as well as less invasive actions like counseling and physiotherapy. This resource is designed for a high-level summarization around the occurrence of a procedure, and not for specific procedure log documentation, which is a concept that does not yet have a defined FHIR Resource.

Imaging

GetStudies (2009)

Queries the following reports or work lists: Status Report, Reading Work List, Signing Work List, Protocol Work List and returns the list of studies that meet certain criteria. You can use this service to maintain the list of studies for a particular reading physician or work as a report to get studies that meet any particular criteria. This service is a synchronous service.

GetStudies (2014)

Queries the following reports or work lists: Status Report, Reading Work List, Signing Work List, Protocol Work List and returns the list of studies that meet certain criteria. You can use this service to maintain the list of studies for a particular reading physician or work as a report to get studies that meet any particular criteria. This service is a synchronous service.

MyChart

ActivatePatient

Enables or disables a patient’s access to MyChart. This service should be used to keep the account status (enabled or disabled) in agreement between the external system and MyChart.

CancelAppointment

Cancels an appointment for a patient. This service is invoked if the patient cancels a future appointment in the external system, or the external system cancels an appointment.

CheckAccessCode

Validates a patient’s MyChart access code, or verifies that an access code exists for the patient. If the access code is provided, this service returns whether the provided code matches the one in the patient’s record. In this case, it also returns whether the code is expired. If the access code is not provided, this service returns whether the patient has a valid, unexpired access code.

ClearAccessCode

Inactivates a patient’s MyChart access code.
CreateAccount

Creates a MyChart account for the given patient, allowing future access to the system. This service would typically be called by an external patient portal application to MyChart that has created a new account in its system, or when a user in the external patient portal has been validated for access to information provided by MyChart.

CreateProxyOnlyAccount

Creates a MyChart account for an existing patient and marks it as a proxy-only account. This is an account (such as a child’s) that cannot log in directly, but can be accessed by another MyChart user (such as a parent) once a relationship is established.

DeleteProxyRelationship

Deletes the proxy relationship between the two patients if one exists. This service would typically be invoked by an external system that is providing administrative staff an alternative to managing proxy information through the Epic Hyperspace application.

GetAlerts

Returns alerts for a MyChart patient. The list of alerts varies, depending upon the list configured for the system and the security available to the patient. Different security settings might apply to proxy access. This service should be invoked after the patient has been authenticated by the external system, or to display alerts to a proxy for a particular patient.

GetAlerts (v2)

Returns alerts for a MyChart patient or for a proxy on behalf of a MyChart patient. The list of alerts varies depending upon the list configured for the system and the security available to the patient. Different security settings might apply to proxy access. This service should be invoked after the patient has been authenticated by the external system, or to display alerts to a proxy for a particular patient.

GetAuthorizedFeatures (v2)

Returns the list of features that a MyChart patient or authorized proxy is authorized to access. This service is typically called when a patient portal application that is external to MyChart displays links to features within MyChart. The list of features that this service returns can be used to dynamically hide or make available links only to those features that the patient is authorized to access. This service should be invoked after the patient has authenticated herself to the patient portal application, but before the patient has access to links to features.

GetCenters

Returns the list of centers where the patient can schedule an appointment. This service would typically be invoked within the appointment scheduling workflow. The service can be used in the scenario where the external system needs to provide the patient with the option of choosing the center he wants to visit for an appointment after choosing the type of visit that he is scheduling.
GetEmailAddress
Returns the primary e-mail address for a MyChart patient. This service would typically be invoked to ensure that the e-mail address on file in MyChart is consistent with the e-mail address maintained by the external system.

GetFutureAppointments
Returns the list of upcoming appointments for a patient along with useful information regarding each of the appointments. This service should be invoked by an external system looking to provide the patient with access to upcoming appointments outside the MyChart application.

GetOpenSlots
Returns the list of time slots in which the caller can schedule an appointment for a patient.

GetPatientsByAccessCode
Validates patient information required for account creation by a single sign-on (SSO) system, and returns the patient ID. The patient ID (and ID type) should then be stored by the SSO system to be used for subsequent Web services calls, such as CreateAccount, GetAlerts (v2), and GetAuthorizedFeatures (v2).

GetPCP
Returns the list of Primary Care Providers (PCPs) assigned to the patient, as well as the departments and locations where the providers work. This service would typically be invoked to get a list of PCPs in order to schedule an appointment for a patient.

GetProviders
Returns the list of providers with whom the patient can schedule an appointment, given a visit type and the optional centers. This service is invoked within the appointment scheduling workflow. After the patient has specified the type of visit he needs to schedule and, optionally, the center in which to schedule the appointment, this service can be called to return the list of providers with whom the patient is able to schedule the given visit type in the specified center.

GetProxyInformation
Returns proxy information for a given MyChart patient. The proxy information has three sections: 1) Patients who have access to the given patient’s MyChart account (Proxies List); 2) Access Codes that can be used to gain access to the given patient’s record (Access Codes List); and 3) Patient accounts to which the given patient has proxy access (Access List). This service would typically be invoked by an external system to provide administrative staff with this information as an alternative to accessing the information through Hyperspace.

GetVisitTypes
Returns the list of visit types that the patient can schedule. This service should be invoked within the appointment scheduling workflow.

ScheduleAppointment
Allows authorized patients to schedule their own appointments. This service would be invoked after the patient is presented with a list of available time slots returned by the GetOpenSlots service.
**SetProxyRelationship**

Sets up new proxy relationships or modifies previously existing proxy relationships. If a proxy relationship exists and this service is invoked, the existing proxy relationship is updated.

**UpdateEmailAddress**

Sets the primary e-mail address for a MyChart patient. This service would be invoked if the external system is collecting the patient’s e-mail address outside of MyChart. This service should be called whenever the external system is updated with a new e-mail address for a patient.

**Orders**

**DiscontinueMedicationOrders**

Discontinues an ambulatory medication order. The service verifies that the user being passed in has the appropriate security to discontinue the order. The service also verifies that the order has not been previously discontinued. The patient associated with the order must be the same as the patient whose records are passed in through the service. The service requires a reason for discontinuation.

**PlaceOrdersUsingTemplate**

Allows pended orders to be placed in an ambulatory context using templates. For example, physicians can build templates to capture the metadata necessary to write orders, and order transmittal rules determine the routing of orders. The orders must be signed within EpicCare, which activates automated order validation, diagnosis associations, and interaction alerts. Then the order is routed based on transmittal rules. Several orders can be placed for a single patient using this service. An existing patient encounter can be specified in the service, or the service can be used to create a new patient encounter where all orders that are part of the request are part of the newly created encounter. The service returns the order identifiers created as part of this call or any errors that were generated as part of order creation. It returns the encounter identifier if a new patient encounter was created.

**Patient Access**

**ClearAccessCode**

Clears a patient’s MyChart access code.

**CreateNonPatientWebAccount**

Creates a non-patient MyChart account, often used by parents of patients at a pediatric organization.

**CreatePatientWebAccount**

Creates a MyChart account for an existing patient.

**eCheckInStatus**

Updates the status of online check-in. This service allows individual steps to be updated such as demographics verification, as well as the overall status of check-in. Additionally, it allows supplementary co-pay data to be collected.
**GetAlerts**

Returns alerts for a MyChart patient or for a proxy on behalf of a MyChart patient. The list of alerts varies, depending upon the list configured for the system and the security available to the patient. Different security settings might apply to proxy access. This service should be invoked after the patient has been authenticated by the external system, or to display alerts to a proxy for a particular patient.

**GetCurrentHealthIssues (2014)**

Returns the list of current health issues for a patient, as well as ICD codes and code sets. You might use the ICD codes for content linking. The list of current health issues returned by this service matches what is displayed in MyChart.

**GetEmailAddress**

Returns the primary e-mail address for a MyChart patient. This service is typically invoked to ensure that the e-mail address on file in Epic is consistent with the e-mail address maintained by the external system.

**GetFutureAppointments**

Returns information about future appointments, including information needed for online check-in such as check-in status, and additional information about the upcoming appointment such as department address and provider photo URL.

**GetHistoricalTestResults**

Returns historical test results available to MyChart by component records.

**GetMyChartSignupPatient**

This web service is intended to be used by a custom patient portal signup webpage during in-clinic patient portal signup. Hyperspace will launch a web browser including two query parameters, InfoID and Client. This web service will take those two query parameters as inputs and will return a set of patient IDs. The IDs can then be used with other web services to activate a patient for the patient portal. The web site launched during in-clinic signup is can be configured on the MyChart Access in Hyperspace Options page in the Login and Access Configuration section of the MyChart system definitions.

**GetNonPatientDemographics**

Returns demographics information for a non-patient account.

**GetPatientBalance**

Returns billing information, including the patient’s account IDs and outstanding balances. For example, if you use MyChart with a third-party billing vendor system, you can pass appropriate information to your billing vendor.

**GetPatientByAccessCode**

Validates patient information required for account creation by a single sign-on (SSO) system, and returns the patient ID. The patient ID (and ID type) should then be stored by the SSO system to be used for subsequent Web services calls, such as CreateAccount, GetAlerts, and GetAuthorizedFeatures.
GetPatientDemographics
Returns patient demographics in a way that’s similar to how demographics are retrieved from MyChart.

GetQuestionnaireList
Returns the list of questionnaires available to a patient in MyChart. This service is proxy-aware, and will enforce proxy access and MyChart security rules.

GetScheduleDaysForProvider
Returns a list of "schedule days" for a provider within the given range. "Schedule days" represents a single day and then lists the openings found for the provider. If there are no slots listed, then this can be interpreted as affirmation that there are no slots. The search will not quit early, it will provide all openings for all days within the range.

GetTestResultDetails
Returns details of a test results for a patient including imaging addenda and scans.

PostPayment
Posts a payment to a billing account. This service can create a MyChart account if the patient is not MyChart active.

ScheduleAppointmentWithInsurance
Checks that a given provider is enabled for open scheduling, the department is configured for open scheduling, and the visit type is enabled for open scheduling in the given department.

SetLoginID
Sets the login ID for an existing MyChart account. You can use this service for patient and non-patient MyChart accounts.

SubmitInsurance
Submits patient’s insurance information.

UpdateNonPatientDemographics
Updates demographics for a non-patient account.

UpdatePatientDemographics
Updates patient demographics in a way that’s similar to how it’s done in MyChart.

Patient Access – Mobile

AddFlowsheetReadings
Adds new readings to a patient-entered flowsheet, or edits or deletes existing flowsheets.

AuthenticateWebAccount
Authenticates a MyChart user, given a username and password.
CancelAppointment
Cancels upcoming appointments for a patient.

ConfirmAppointment
Confirms an upcoming appointment for a patient.

DeleteMessage
Deletes a given set of messages.

GetAlerts
Returns alerts for a given patient.

GetAllergies
Returns the list of current allergies for a patient.

GetCancelAppointmentDetails
Returns information about canceling an upcoming appointment for a patient, including whether comments are allowed, whether a cancel reason is required, and whether there are linked appointments.

GetCurrentHealthIssues (2011)
Returns the list of current health issues for a patient.

GetFlowsheetReadings
Returns all of the patient’s readings from within the requested timeframe for a given flowsheet.

GetFlowsheets
Returns a list of all the patient-entered flowsheets assigned to the given patient.

GetFutureAppointments
Returns the list of upcoming appointments for a patient along with useful information regarding each of the appointments.

GetHealthAdvisories
Returns the list of health advisories for a patient.

GetImmunizations
Returns the list of immunizations for a patient.

GetMedicalAdviceRecipients
Returns the list of valid users, pools, and subjects that a given MyChart user can use to send messages for a given patient.
**GetMedications**
Returns the list of current and recently expired outpatient medications for a patient.

**GetMessageDetails**
Returns details for a particular MyChart message.

**GetMessageDetails**
Returns details for a particular MyChart Inbox message, including questionnaire information.

**GetMessages**
Returns the list of inbox messages for a patient.

**GetPastAppointments**
Returns the list of past appointments for a patient. The service returns 25 test results at a time.

**GetPastAppointmentDetails**
Returns the details of a past appointment for a patient.

**GetPatientPhoto**
Returns the photo saved to a patient’s MyChart account or patient record in Hyperspace. This service can also resize the photos returned.

**GetPreferredPharmacies**
Returns the patient’s preferred pharmacies. This service can be used with GetMedications and RequestMedicationRefill to submit medication refill requests.

**GetRefillPaymentInformation**
Returns payment information for medication refill orders.

**GetSchedulingInformation**
Returns general information about overall scheduling options and configuration.

**GetSchedulingProvidersInformation**
Returns a list of possible providers given the specified reason for a visit.

**GetSentMessageDetails**
Allows patients to view a message they previously sent to a Hyperspace user from within your custom application or patient portal. Additionally, you can restrict and resize message attachments to improve performance and ensure that data successfully passes through your web infrastructure.

**GetSentMessages**
Returns the list of messages sent by a patient.
GetSlotReviewInformation
Given a specific slot, returns more detailed information about the slot for the specified patient.

GetSlotsInformation
Returns a list of possible slots.

GetTestResultDetails (Summer 2009 Version)
Returns details of a test result for a patient.

GetTestResultDetails (Epic 2012 Version)
Returns details of a test result, including imaging addenda and scans.

GetTestResults
Returns the list of test results for a patient. The service returns 25 test results at a time.

GetVisitTypeInfo
Returns scheduling information about a given Reason for Visit index.

LogAudit
Logs audit information for a patient.

PostAppointment
Attempts to schedule an appointment from the given slot information.

RemoveMyChartPatientPhoto
Allows patients and patient proxies to remove patient photos from your custom mobile application or patient portal.

RequestMedicationRefill
Requests a MyChart medication refill. A Refill Authorization In Basket message can be sent for the refill, or it can be automatically refilled if the pharmacy is integrated with Willow.

SendAppointmentCancellationRequest
Sends a message to the organization requesting that an appointment be cancelled.

SendMessage
Sends a message to the patient. This is used for both new messages as well as message replies.

SendMessage
Sends a message with attachments from the patient. Can be used for both new messages and message replies.
**SetMyChartPatientPhoto**

Allows patients and patient proxies to choose and upload a patient photo for your custom mobile application or patient portal. Depending on your configuration, the photo can also be immediately saved to the patient’s chart and visible in Hyperspace. Alternatively, you can require a Hyperspace user to go to the Demographics activity or Appointment Desk to review and approve the photo before it is saved to the patient’s chart and visible to clinicians.

**Pharmacy**

**CancelFills**

Allows for communication between a third-party Point of Sale (POS) pharmacy system and Willow Ambulatory. With this service, pharmacists and technicians can perform a fill cancellation or return workflow by taking action in the Point of Sale system. This web service requests that the specified fills be canceled in the pharmacy system. If you use Willow Inventory, this service handles updating inventory levels and reversing any charges that have already been processed for the fill.

**DispenseFills**

Requests that specified fills be marked as dispensed in the pharmacy system.

**GetCreditCards**

Returns the credit cards for a given patient that are available for use at a given pharmacy. This web service can also provide an IVR system the ability to query what credit cards are available on a patient’s record that can be charged for a fill. When requesting a refill, a patient can then specify which credit card from their record to use for the payment for the fill.

**GetPrescriptionInfo**

Returns prescription information, including information about the patient who owns the prescriptions.

**GetPrescriptionInfoByld**

Returns prescription information, including information about the patient who owns the prescriptions.

**GetVersion**

Retrieves the current version of a Web service.

**RequestFills**

Requests that the pharmacy system initiates a fill for the specified prescriptions.

**Scheduling**

**AddOnShiftProviderSchedule**

Adds shift provider schedule information, which includes shift start date and time, shift end date and time, department in which the provider is scheduled, and role. Managers and charge nurses responsible for making clinician-to-patient assignments can use this information to perform their assignment workflows without having to reference scheduling information from outside of the system.
AssignReferralToAppointment
Assigns or un-assigns a referral to an appointment. Both the referral and appointment must already exist.

CancelAppointment
Marks an appointment as canceled.

EditAppointmentNotes
Updates appointment notes for a single appointment, overwriting any previous appointment notes.

EditProviderSlot
Edits pre-existing provider slots. This service can't create new slots.

GetAutoSchedulerSolutions
Returns available solutions that can be scheduled as appointments.

GetAutoSchedulerSolutionsAdvanced
Returns available solutions that can be scheduled as appointments. This service has some advanced-use parameters that GetAutoSchedulerSolutions does not.

GetDepartmentWaitTimes
Returns average wait times for one or more departments. This time is calculated based how long it has historically taken patients to be seen.

GetOpenSlots
Returns available slots that can be booked as appointments.

GetPatientAppointments
Returns a list of the appointments for a patient.

GetPatientNotificationPreferences
Returns a specific patient’s notification preferences, including email and notification options.

GetPatientSchedulableOrders
Returns orders that are schedulable for a given patient.

GetPatientSchedulingPreferences
Returns a specific patient’s scheduling preferences, including location, days of the week, provider sex, provider language, and time of day.

GetPatientWaitList
Returns the wait list entries that match and are allowed to be viewed by a user.

GetProviderAppointments
Returns the appointments for a provider or a list of providers.
GetProviderAvailability
Returns the schedule availability for a provider for a range of dates.

GetProviderSchedule
Returns the schedule information for a provider on a given date.

GetScheduleInstructions
Returns scheduling instructions.

PostAppointmentPayments
Posts payments to Epic for individual appointments. This web service does not process the payment or create appointments.

ScheduleAppointment
Schedules an appointment.

SetInterpreterFlag
Sets the interpreter flag on an appointment to Not Needed or Needs Review.

SetPatientNotificationPreferences
Sets a patient’s notification preferences, including email and notification options.

SetPatientSchedulingProcesses
Sets a patient’s scheduling preferences, including location, days of the week, provider sex, provider language, and time of day.

UpdateWaitListEntry
Creates a new wait list entry for a patient or updates an existing wait list entry for a patient.

Security

AcceptBreakTheGlass
Logs an accepted Break-the-Glass form to run through the action lists set up in Epic.

ActivateUser
Activates a user record, unblocks a blocked user, clears an end date set in the past, updates a record based on linked templates, and adds a comment to the login history if necessary to prevent premature automatic inactivation.

CancelBreakTheGlass
Logs cancelled Break-the-Glass prompts.

CheckBreakTheGlass
Runs the released checks in Break-the-Glass to determine the type of access the user can have.
**CreateUser**

Creates a user record and returns the ID of the new record.

**CreateUser2**

Creates a user record, populates the initial items, and enables you to add multiple linkable templates, employee demographics, and category report groupers.

**DeleteUser**

Soft-deletes a user record. In an Intraconnect community, this service can be used only on the owning deployment.

**ForcePasswordChange**

Sets a flag that forces a user to select a new password on next login. It should be called right after setting a user’s password if the password change was done by an administrator.

**InactivateUser**

Inactivates a user record.

**InitializeBreakTheGlass**

Returns Break-the-Glass initialization information, including data requirements for the reason and explanation fields, the legal message, a list of possible reasons, the message to display for inappropriate access, the default Hyperspace timeout in minutes, and any reason-specific overrides for the explanation field’s data requirement.

**SetDutchName**

Updates a user’s name according to the default rules for the Dutch locale.

**SetReportSelectionCriteria**

Updates the Report Selection Criteria items for an existing user record.

**SetUserExternalPasswords**

Sets passwords associated with a user’s external IDs. A password cannot be set for an external ID type until an ID has first been set for that type.

**SetUserPassword**

Sets the user’s Epic password to a new value, and updates the user’s password history. This service does not allow setting a password to a value that violates the environment’s password complexity requirements, and it can be used only to change a password for users using Epic native authentication.

**UpdateCommunityUser**

Allows web service write access to EpicCare Link, PlanLink, and Healthy Planet Link user records.
**UpdateUser**

Updates user information for a specified user. This service cannot be used to set items that are controlled by a template.

**UpdateUser2**

Updates user information for a specified user. This service cannot be used to set items that are controlled by a template. This service expands on the original UpdateUser and allows you to modify multiple linkable templates, employee demographics, and category report groupers for existing users.

**ViewCommunityUser**

Allows web service read access to EpicCare Link, PlanLink, and Healthy Planet Link user records.

**ViewUser**

Enables you to view items in a user record.

**ViewUser2**

Returns the values of the fields that can be set using the CreateUser2 or UpdateUser2 web services.

**Specialty**

**GetPatientResultsbyProcedure**

Returns result information for a particular procedure grouped by order. This service is similar to the GetPatientResultComponents service, but is designed for searches by procedures, such as an X-Ray.

**GetSurgicalRecord**

Returns surgical case and case log information. If both case and log records exist, data is returned from both of the master files.

**GetSurgicalRecord (2014)**

Returns surgical case and case log information. If both case and log records exist, data is returned from both of the master files. This service also logs user information to the audit trail.

**GetSurgicalRecord (2015)**

Returns surgical case and case log information. If both case and log records exist, data is returned from both of the master files. This service also logs user information to the audit trail.

**GetSurgicalRecords**

Returns surgical case and case log information for all the surgical records selected based upon the filtering criteria provided in the service request. If both case and log records exist, data is returned from both of the master files.
GetSurgicalRecords (2014)
Returns surgical case and case log information for all the surgical records selected based upon the filtering criteria provided in the service request. If both case and log records exist, data is returned from both of the master files. This service also logs user information to the audit trail.

GetSurgicalRecords (2015)
Returns surgical case and case log information for all the surgical records selected based upon the filtering criteria provided in the service request. If both case and log records exist, data is returned from both of the master files. This service also logs user information to the audit trail.

GetWaitTimes
Returns average wait times for a given department.