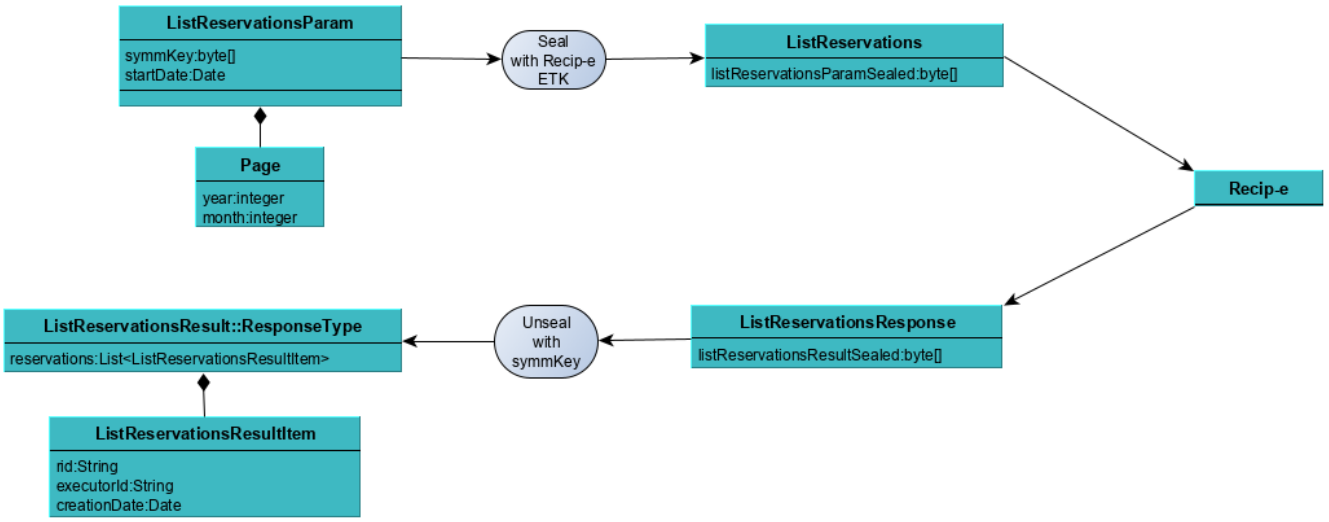
Overview operations

1. LISTRESERVATIONS

Operation Name	***ListReservations(ListReservations)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This function retrieves the reserved Recip-e identifiers and their reservation dates. The request contains a start date: only reservations after the start date are returned. The SDK maintains a list of previous returned reservations (RID + reservation date). The status of the prescription is not modified by calling this operation.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	ListReservationsParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		startDate	*String*	*The start date indicating the beginning of the period of the requested reservations.*
		page(optional)	*Page*	*The page to return (by default: the current month).*
		Session(optional)	*Session*	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on the server of subsequent listrequests by storing operation specific state.*
	Page	*Month(unused)*	*int*	*The month of interest*
		year(unused)	*int*	*The year of interest*
		pageNumber	*int*	*The pagination number*
	Session		*byte []*	*The session context. This object is unreadable for the client.*
	ListReservations	*ListReservationsParamSealed*	*Byte[]*	*The ListReservationsParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	ListReservationsResponse	*ListReservationsResultSealed*	*Byte []*	*Sealed byte representation of the ListReservationsResult-object. Unseal with the symmetric key from the request-parameters.*
	ListReservationsResult	*status*	*StatusType*	*Contains the StatusType-object - object*
		items	*List*	*List of ListReservationsResultItem-objects*

	hasMoreResults	Boolean	*Flag indicating if there are more results*
	session	Session	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on the server of subsequent listrequests by storing operation specific state.*
ListReservationsResultItem	*rid*	*String*	*The Recip-e identifier*
	executorId	*String*	*The executor identifier*
	creationDateReservation	*Calendar*	*The creationdate of the reservation*
	lastUpdateReservation	*Calendar*	*The lastupdate of the reservation*
	Prescription	*GetOpenPrescriptionForExecutor*	*The prescription without the proof*
	creationDate	*Calendar*	*The creationdate of the prescription*
	expirationDate	*Calendar*	*The expirationdate of the prescription*
	prescriptionStatus	PrescriptionStatus	The prescription status
	patientId	*String*	*The patient identifier*
	prescriberId	*String*	*The prescriber identifier*
	emailAddress	*String*	*The email address*
	telephoneNumber	*String*	*The telephone number*
	contactPreference	*contactPreference*	*The preferred contact. (possible values are defined in the recipe-core.xsd file under the simpleType: contactPreference)*
GetOpenPrescriptionForExecutor	*rid*	*String*	*The Recip-e identifier*
	prescriptionStatus	*PrescriptionStatus*	*The prescription status of the Recip-e identifier*
	creationdate	*Date*	*The creation date of the prescription*
	patientId	*String*	*The patientId*
	prescription	*Byte []*	*The prescription bytes*
	feedbackAllowed	*Boolean*	*Flag indicating feedback is allowed or not*
	expirationDate	*Calendar*	*The expiration date of the prescription*
	prescriptionType	*String*	*The prescription type*
	encryptionKeyId	*String*	*The encryption key id*
	prescriberId	*String*	*The prescriber id*
StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*

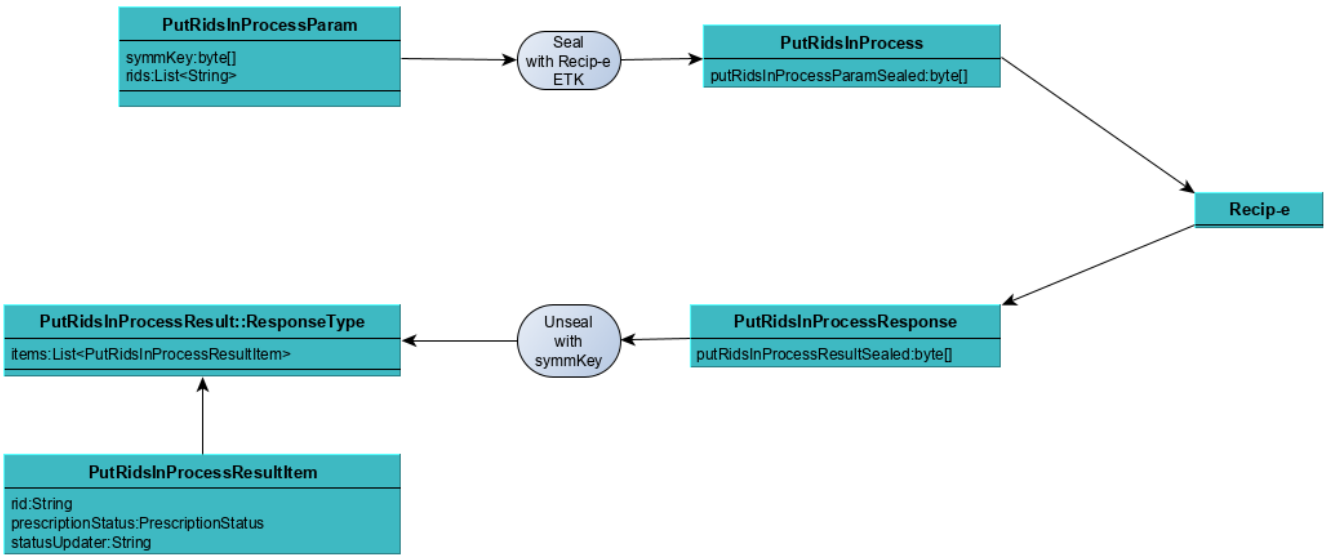
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



2. PUTRIDSINPROCESS

Operation Name	***PutRidsInProcess(PutRidsInProcess)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This function sets a list of prescriptions in status InProcess. The request is limited to process 10 RIDs in one request. The response returns the set of RIDs and their actual status. If the status in the response is not the expected InProcess-status, this means that a problem occurred to update the RID to this status.* *An example response could be:* *{ (RID1, InProcess, statusUpdater1), (RID2, Archived, statusUpdater2) }* • *As RID1 has the status InProcess, this update took place correctly. The statusUpdater is the person who called the PutRidsInProcess-operation.* • *As RID2 has the status Archived, the update to status InProcess could not take place and the operation will return the actual status from the prescription. The statusUpdater is the person who archived the prescription.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	PutRidsInProcessParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		rids	*List<String>*	*The Recip-e identifiers to put in process*
	PutRidsInProcess	*PutRidsInProcessParamSealed*	*Byte[]*	*The PutRidsInProcessParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*

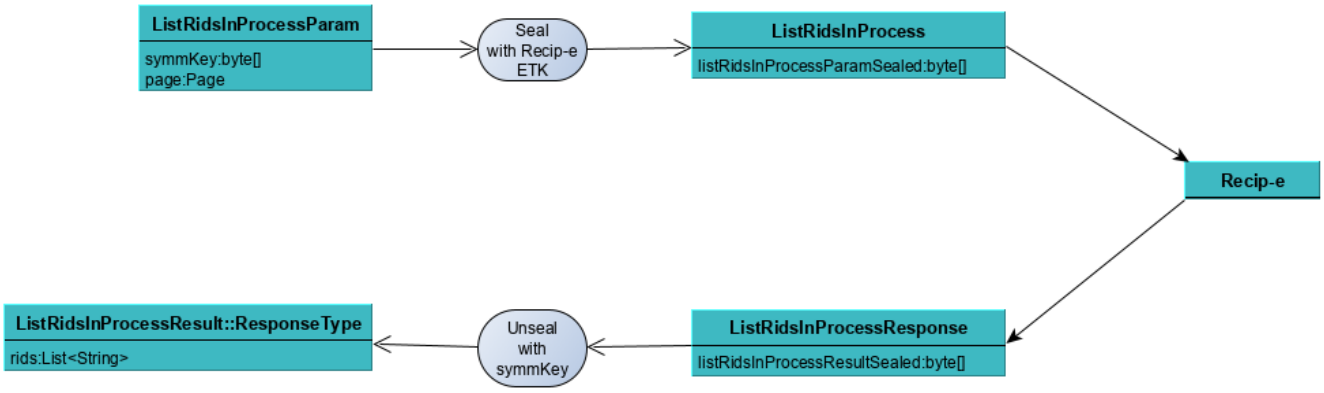
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	PutRidsInProcessResponse	*putRidsInProcessResultSealed*	*Byte []*	*Sealed byte representation of PutRidsInProcessResult. Unseal with the symmetric key from the PutRidsInProcessParam-*
	PutRidsInProcessResult	*Status*	*StatusType*	*Contains the StatusType-object - object*
		items	*List*	*List containing items of type the PutRidsInProcessResultItem-object*
	PutRidsInProcessResultItem	*rid*	*String*	*The Recip-e identifier*
		prescriptionStatus	*PrescriptionStatus*	*The prescription status*
		statusUpdater	*String*	*The statusUpdater*
	StatusType	*statusCode*	*String*	*100: ok 300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



3. LISTRIDSINPROCESS

Operation Name	***ListRidsInProcess(ListRidsInProcess)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This function retrieves a list of RIDs which have the status InProcess. The status of the prescription is not modified by executing this function.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	ListRidsInProcessParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		page(optional)	*Page*	*The page to return. (by default: the last previous 12 months).*
		page(optional)	*Page*	*The page to return. (by default: the last previous 12 months).*
		Session(optional)	*Session*	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on the server of subsequent listrequests by storing operation specific state.*
	Page	*Month(unused)*	*int*	*The month of interest*
		year(unused)	*int*	*The year of interest*
		pageNumber	*int*	*The pagination number*
	Session		*byte []*	*The session context. This object is unreadable for the client.*
	ListRidsInProcess	*ListRidsInProcessParamSealed*	*Byte[]*	*The ListRidsInProcessParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*

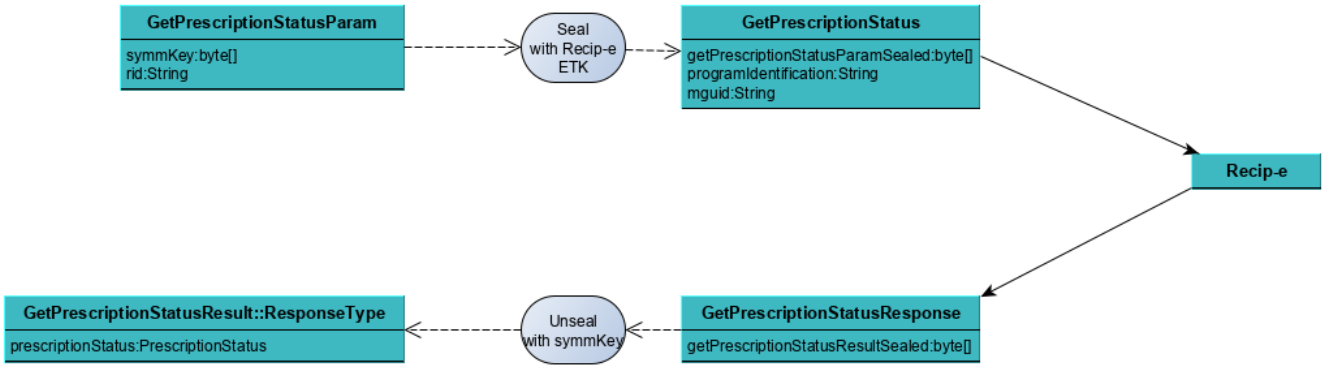
Output Integration Module	Response-entity	Attribute	Type	Description
	ListRidsInProcessResponse	*ListRidsInProcessResultSealed*	*Byte []*	*Sealed byte representation of the ListRidsInProcessResult-object. Unseal with the symmetric key from the GetAll RidInProcessParam-object.*
	ListRidsInProcessResult	*status*	*StatusType*	*Contains the StatusType-object*
		rids	*List*	*List of Recip-e identifiers which have status <InProcess>*
		hasMoreResults	Boolean	*Flag indicating if there are more results*
		session	Session	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on the server of subsequent listrequests by storing operation specific state*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error validation x,y,z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



4. GETPRESCRIPTIONSTATUS

Operation Name	***GetPrescriptionStatus(GetPrescriptionStatus)***
Class Name	***ExecutorIntegrationModuleV4***

Description	*This function retrieves the status of one prescription.This operation does not modify the status of the prescription.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	GetPrescriptionStatusParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		rid	*String*	*The rid or ridEx*
	GetPrescriptionStatus	*getPrescriptionStatusParamSealed*	*Byte[]*	*GetPrescriptionStatusParam sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	GetPrescriptionStatusResponse	*getPrescriptionStatusResultSealed*	*Byte []*	*Sealed byte representation of GetPrescriptionStatusResult.*
	GetPrescriptionStatusResult	*status*	*StatusType*	*Contains the StatusType-object*
		prescriptionStatus	*PrescriptionStatus*	*The prescription status*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message* *e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*

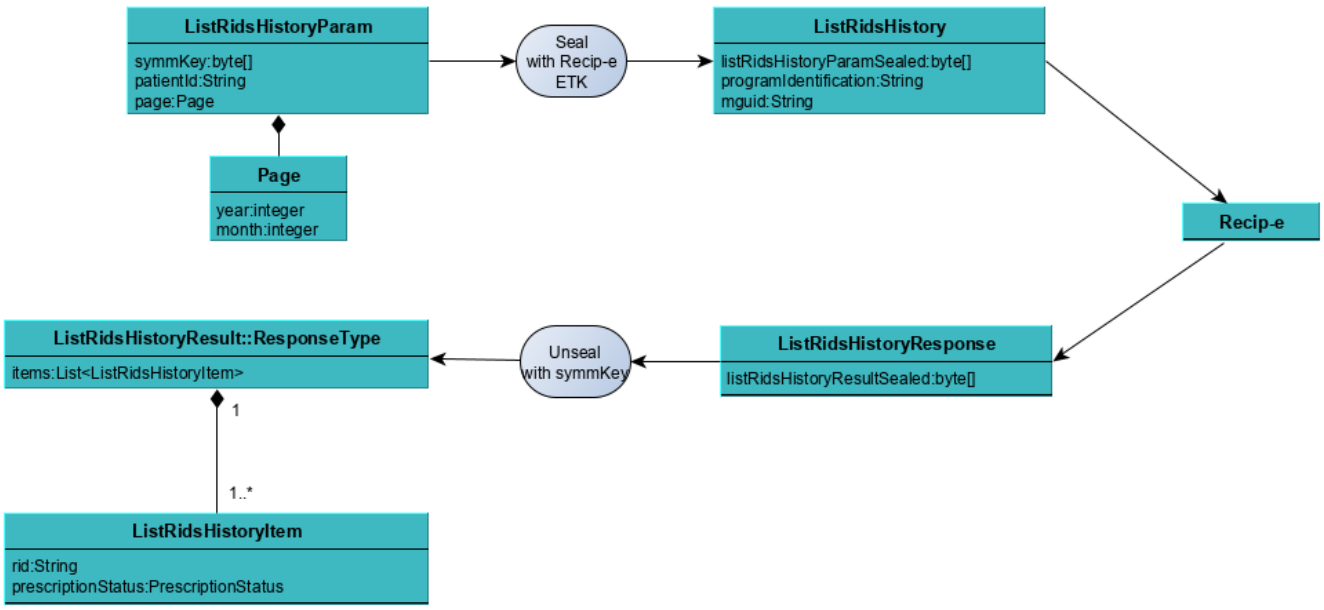


5. LISTRIDSHISTORY

Operation Name	***ListRidsHistory(ListRidsHistory)***
Class Name	***ExecutorIntegrationModuleV4***

Description	*This function returns a list of RIDs and their actual status for a given patientId. The statuses of the prescriptions are not modified by calling this operation.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	ListRidsHistoryParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		patientId	*String*	*The patient identifier*
		activeResults	*Boolean*	*Flag indicating to return active or historical prescriptions*
		breakTheGlass	*breakTheGlass*	*breakTheGlass type is defined in UnsealedExecutorServicesV4_schema1*
		page(optional)	*Page*	*The page to return. (by default: the last previous 12 months).*
		Session(optional)	*Session*	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on the server of subsequent listrequests by storing operation specific state.*
	Page	*Month(unused)*	*int*	*The month of interest*
		year(unused)	*int*	*The year of interest*
		pageNumber	*int*	*The pagination number*
	Session		*byte []*	*The session context. This object is unreadable for the client.*
	ListRidsHistory	*ListRidsHistoryParamSealed*	*Byte[]*	*The ListRidsHistoryParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	ListRidsHistoryResponse	*ListRidsHistoryResultSealed*	*Byte []*	*Sealed byte representation of the ListRidsHistoryResult-object. Unseal with the symmetric key from the request.*
	ListRidsHistoryResult	*status*	*StatusType*	*Contains the StatusType-object*
		items	List	*The list of the ListRidsHistoryItem-object*

		hasMoreResults	Boolean	*Flag indicating if there are more results*
		session	Session	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on the server of subsequent listrequests by storing operation*
	ListRidsHistoryItem	*rid*	*String*	*The Recip-e identifier*
		prescriptionStatus	*PrescriptionStatus*	*The prescription status*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*

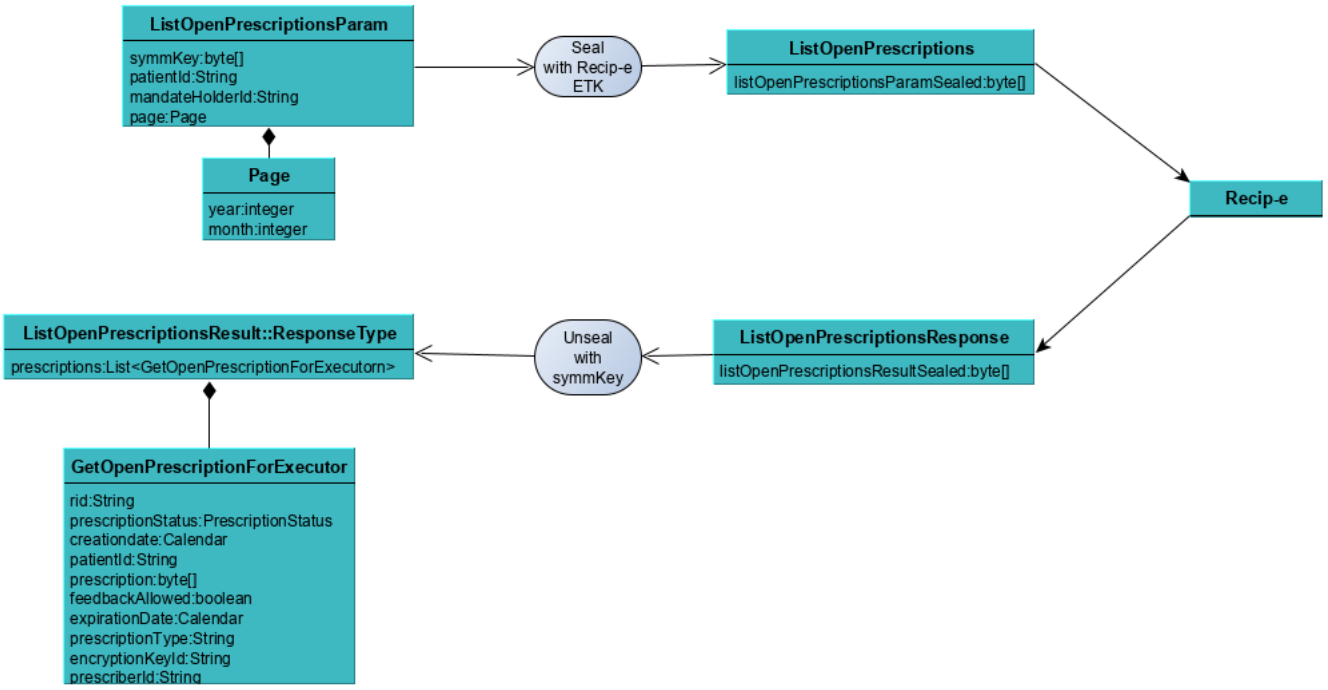


6. LISTOPENPRESCRIPTIONS

Operation Name	***ListOpenPrescriptions(ListOpenPrescriptions)***
Class Name	***ExecutorIntegrationModuleV4***
Description	*This function retrieves a list of prescriptions including the encryption key without the proof. The statuses of the prescriptions are not modified by calling this operation.*
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message.* *Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*

Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	ListOpenPrescriptionsParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		mandateHolderId	*String*	*The mandateholder identifier*
		patientId	*String*	*The patient identifier*
		breakTheGlass	*breakTheGlass*	*breakTheGlass type is defined in UnsealedExecutorServicesV4_schema1*
		page(optional)	*Page*	*The page to return. (by default: the last previous 12 months).*
		session(optional)	*Session*	*The session is returned by the server and is contained in the first paging response. This object*
	Page	*Month(unused)*	*int*	*The month of interest*
		year(unused)	*int*	*The year of interest*
		pageNumber	*int*	*The pagination number*
	Session		*byte []*	*The session context. This object is unreadable for the client.*
	ListOpenPrescriptions	*ListOpenPrescriptionsParamSealed*	*Byte[]*	*The ListOpenPrescriptionsParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	ListOpenPrescriptionsResponse	*ListOpenPrescriptionsResultSealed*	*Byte []*	*Sealed byte representation of the ListOpenPrescriptionsResult-object. Unseal with the symmetric key from the request-parameters.*
	ListOpenPrescriptionsResult	*status*	*StatusType*	*Contains the StatusType-object*
		prescriptions	*List*	*The list of GetOpenPrescriptionForExecutor-objects*
		hasMoreResults	Boolean	*Flag indicating if there are more results*
		session	Session	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on*

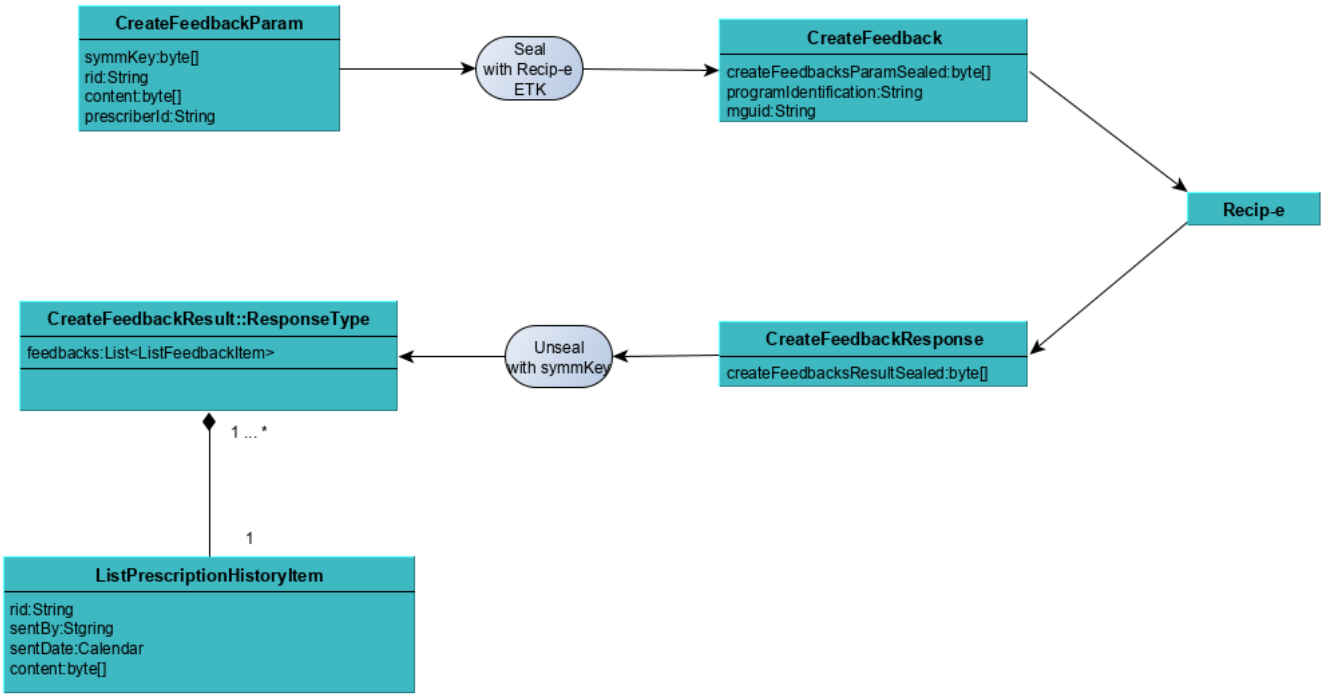
	GetOpenPrescriptionForExecutor	*rid*	*String*	*The Recip-e identifier*
		prescriptionStatus	*PrescriptionStatus*	*The prescription status of the Recip-e identifier*
		creationdate	*Date*	*The creation date of the prescription*
		patientId	*String*	*The patientId*
		prescription	*Byte []*	*The prescription bytes*
		feedbackAllowed	*Boolean*	*Flag indicating feedback is allowed or not*
		expirationDate	*Calendar*	*The expiration date of the prescription*
		prescriptionType	*String*	*The prescription type*
		encryptionKeyId	*String*	*The encryption key id*
		prescriberId	*String*	*The prescriber id*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



7. CREATEFEEDBACK

Operation Name	***CreateFeedback(CreateFeedback)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This operation sends a feedback from an executor to a prescriber.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	CreateFeedbackParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the*
		rid	*String*	*The Recip-e identifier*
		content	*Byte []*	*The content of the feedback*
		prescriberId	*String*	*The prescriber identification*
	CreateFeedback	*createFeedbackParamSealed*	*Byte[]*	*The CreateFeedbackParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	CreateFeedbackResponse	*createFeedbackResultSealed*	*Byte []*	*Sealed byte representation of the CreateFeedbackResult-object.* *Unseal with the symmetric key*

	CreateFeedbackResult	*status*	*StatusType*	*Contains the StatusType-object*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g.*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*

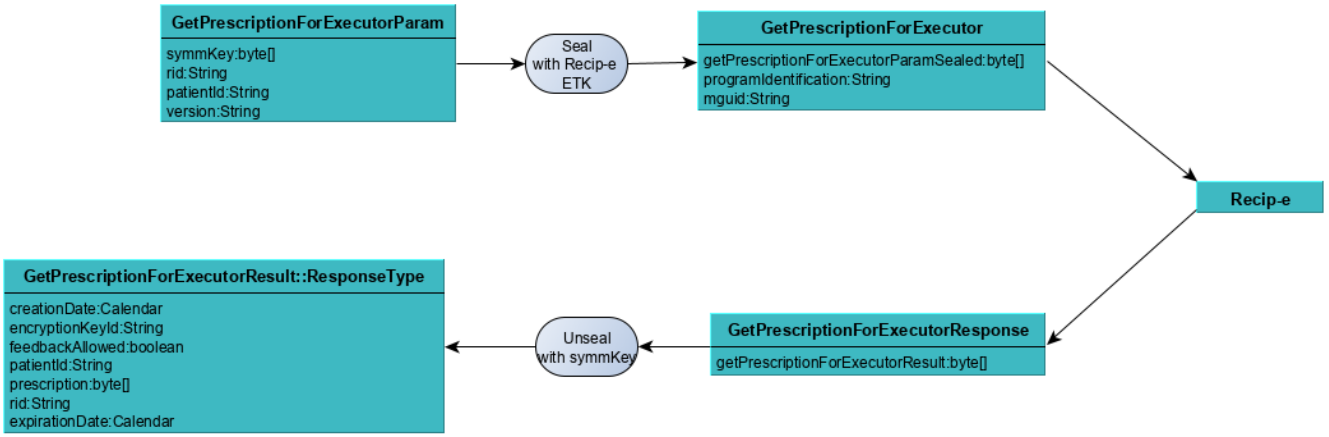


8. GETPRESCRIPTIONFOREXECUTOR

Operation Name	***GetPrescriptionForExecutor(GetPrescriptionForExecutor)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This operation returns the prescription (with encrypted prescription bytes) for a given Recip-e identifier. This operation does modify the status of the prescription to InProcess.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	GetPrescriptionForExecutorParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*

		rid	*String*	The Recip-e identifier
		patientId	*String*	*The patient identifier*
		version	*String*	*The prescription version*
	GetPrescriptionForExecutor	*getPrescriptionForExecutorParamSealed*	*Byte[]*	*The GetPrescriptionForExecutorParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	GetPrescriptionForExecutorResponse	*getPrescriptionForExecutorResult*	*GetPrescriptionForExecutorResultWrapper*	*The GetPrescriptionForExecutorResponse-object*
	GetPrescriptionForExecutorResultWrapper	*getPrescriptionForExecutorUnsealedResultPart*	*GetPrescriptionForExecutorUnsealedResultPa*	*The GetPrescriptionForExecutorUnsealedResultPart-object*
		getPrescriptionForExecutorResultSealed	*Byte []*	*Sealed representation of the GetPrescriptionForExecutorResultSealed-object*
	GetPrescriptionForExecutorUnsealedResultPart	*patientId*	*String*	*The patient identifier*
		prescriptionType	*String*	*The prescription type*
	GetPrescriptionForExecutorResultSealed	*rid*	*String*	*The Recip-e identifier*
		creationDate	*Calendar*	*The creation date*
		patientId	*String*	*The patient identifier*
		feedbackAllowed	*boolean*	*Flag indicating if feedback is allowed*
		prescription	*Byte []*	*The encrypted prescription. Use the encryptionKeyId to decrypt the bytes*
		prescriptionType	*String*	*The prescription type (P0/P1)*
		encryptionKeyId	*String*	*The encryption key identifier in the KGSS register*
		timestampingId	*String*	*The timestamping identifier*
		prescriberId	*String*	*The presciber identifier*
		expirationDate	*Calendar*	*The expiration date of the prescription*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*

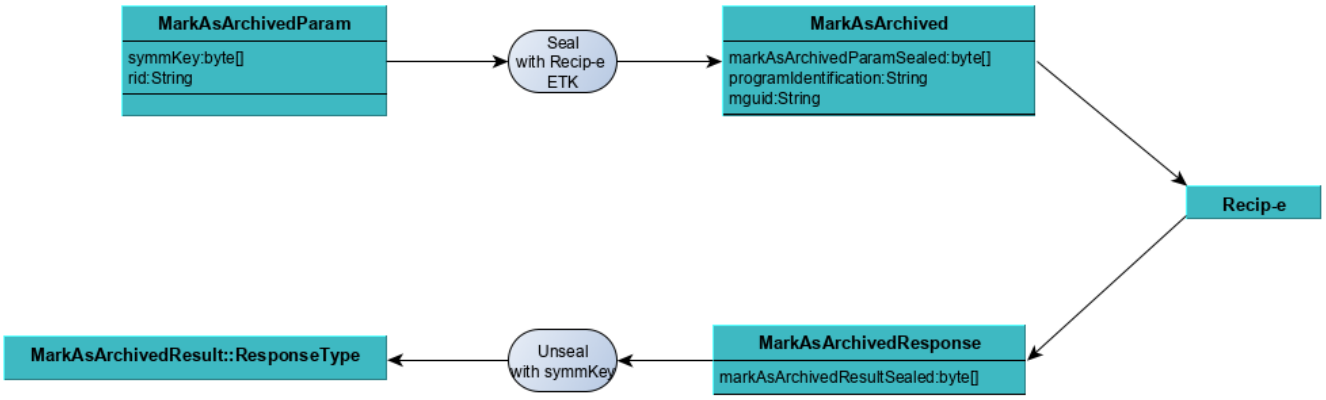
message	*String*	*The error message*
statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



9. MARKASARCHIVED

Operation Name	***MarkAsArchived(MarkAsArchived)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This operation marks the status of a prescription as Archived. The prescription bytes are removed.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	MarkAsArchivedParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		rid	*String*	The Recip-e identifier
	MarkAsArchived	*markAsArchivedParamSealed*	*Byte[]*	*The MarkAsArchivedParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*

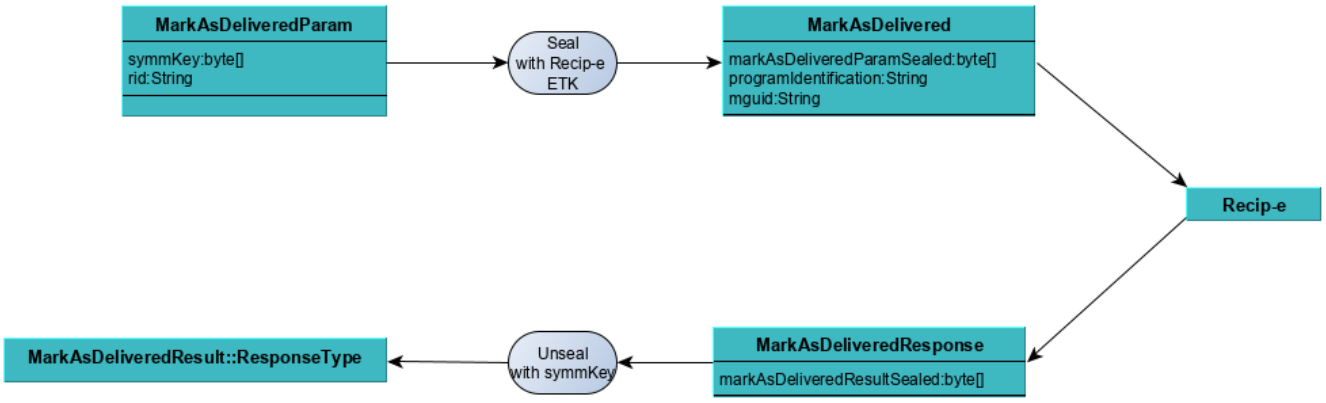
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	MarkAsArchivedResponse	*markAsArchivedResultSealed*	*Byte []*	*Sealed byte representation of the MarkAsArchivedResult-object.*
	MarkAsArchivedResult	*status*	*StatusType*	*Contains the StatusType-object*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



10. MARKASDELIVERED

Operation Name	***MarkAsDelivered(MarkAsDelivered)***
Class Name	***ExecutorIntegrationModuleV4***
Description	*This operation marks status of a prescription as Delivered.*

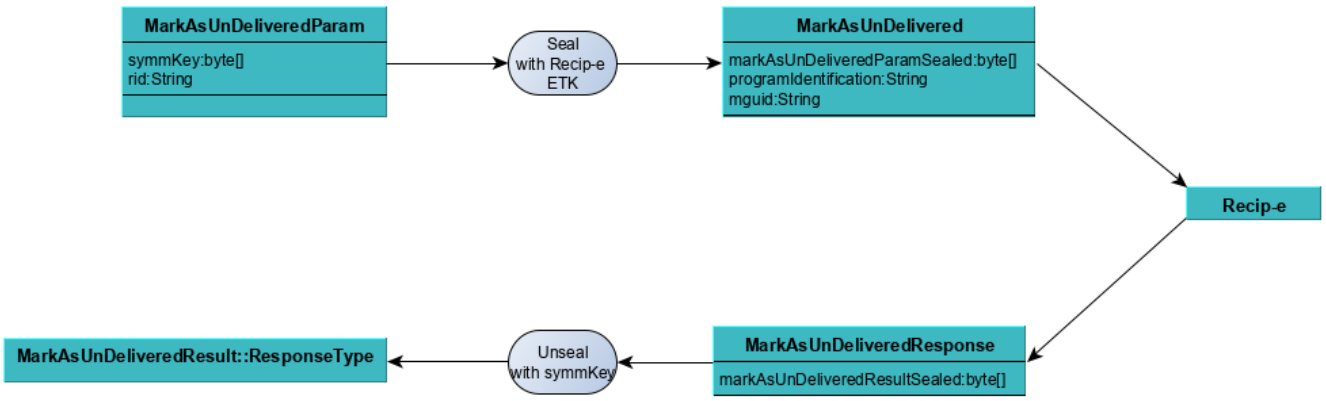
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	MarkAsDeliveredParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		rid	*String*	The Recip-e identifier
	MarkAsDelivered	*markAsDeliveredParamSealed*	*Byte[]*	*The MarkAsDeliveredParam -object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	MarkAsDeliveredResponse	*markAsDeliveredResultSealed*	*Byte []*	*Sealed byte representation of the MarkAsDeliveredResult-object. Unseal with the*
	MarkAsDeliveredResult	*status*	*StatusType*	*Contains the StatusType-object*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



11. MARKASUNDELIVERED

Operation Name	***MarkAsUnDelivered(MarkAsUndelivered)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This operation marks status of a prescription as NotDelivered.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	MarkAsUndeliveredParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		Rid	*String*	The Recip-e identifier
	MarkAsUndelivered	*markAsUndeliveredParamSealed*	*Byte[]*	*The MarkAsUndeliveredParam -object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	MarkAsUnDeliveredResponse	*markAsUnDeliveredResultSealed*	*Byte []*	*Sealed byte representation of The MarkAsUnDeliveredResult-object. Unseal with the symmetric key from the*
	MarkAsUnDeliveredResult	*status*	*StatusType*	*Contains the StatusType-object*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*

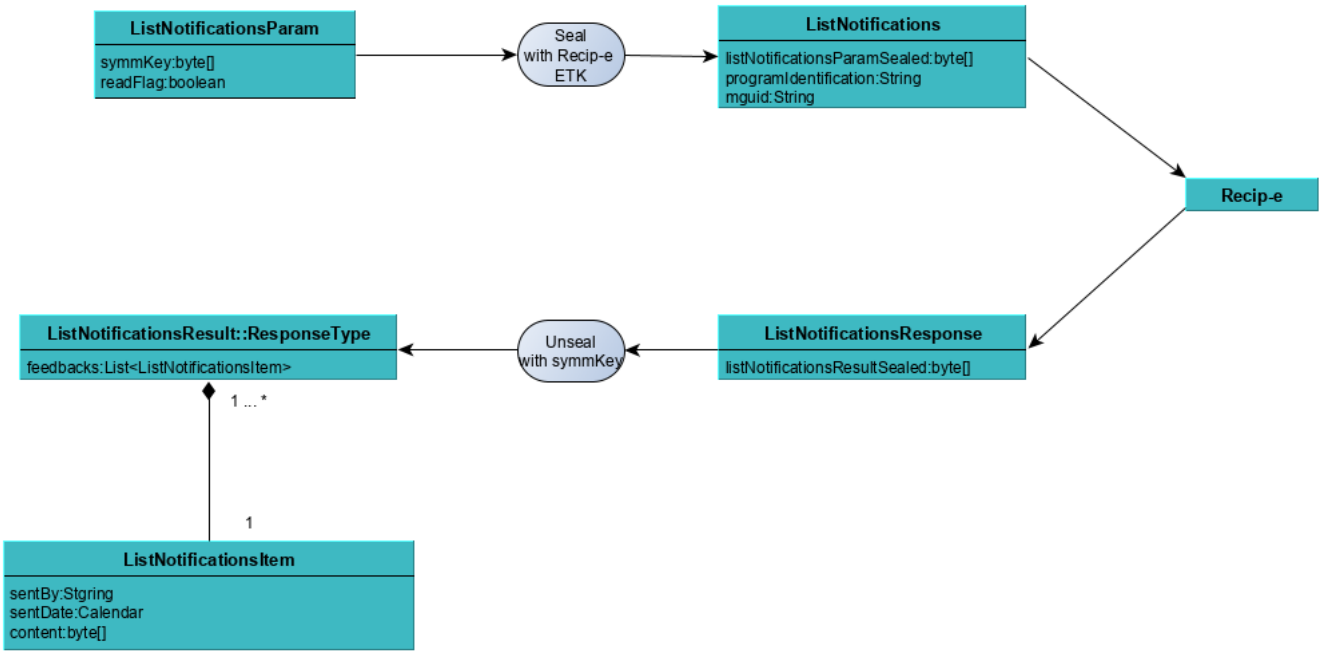
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



12. LISTNOTIFICATIONS

Operation Name	***ListNotifications(ListNotifications)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This operation returns the list of notifications sent b y a prescriber to a given executor.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	ListNotificationsParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		readFlag	*Boolean*	Flag indicating the notification is read or not.

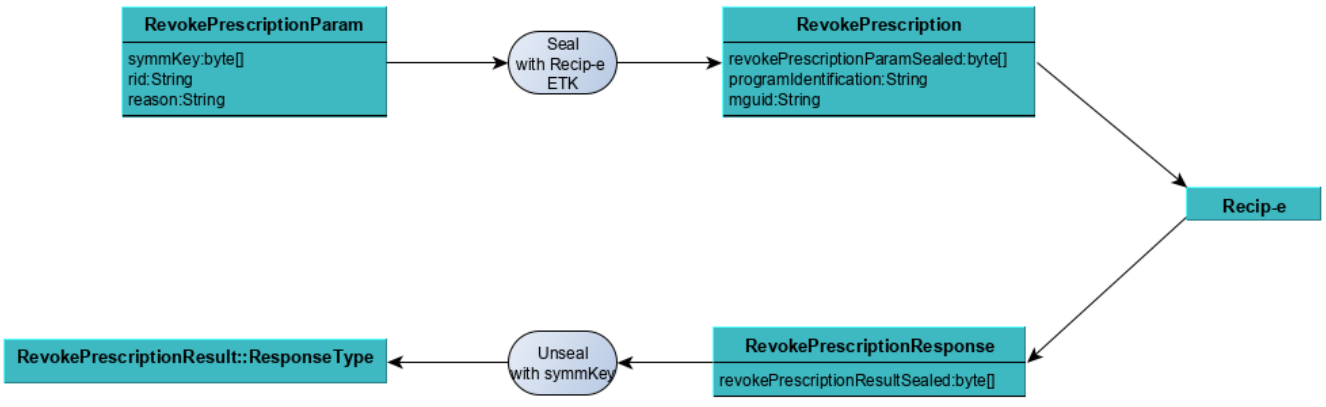
	ListNotifications	*listNotificationsParamSealed*	*Byte[]*	*The ListNotificationsParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	ListNotificationsResponse	*listNotificationsResultSealed*	*Byte []*	*Sealed byte representation of The MarkAsUnDeliveredResult-object. Unseal with the*
	ListNotificationsResult	*status*	*StatusType*	*Contains the StatusType-object*
		notifications	*List*	*List of NotificaitonItem-objects*
	ListNotificationsItem	*sentBy*	*String*	*The identifier of the sender*
		sentDate	*Calendar*	*The sent date*
		content	*Byte []*	*The content bytes*
	StatusType	*statusCode*	*String*	*100: ok 300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



13. REVOKEPRESCRIPTION

Operation Name	***RevokePrescription(RevokePrescription)***			
Class Name	***ExecutorIntegrationModuleV4***			
Description	*This operation updates the status of a prescription to Revoked. The prescription bytes are removed.*			
Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	RevokePrescriptionParam	*symmKey*	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		rid	*String*	The Recip-e identifier
		reason	*String*	The reason of revocation
	RevokePrescription	*revokePrescriptionParamSealed*	*Byte[]*	*The RevokePrescriptionParam-object sealed with the Recip-e ETK*

		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	RevokePrescriptionResponse	*revokePrescriptionResultSealed*	*Byte []*	*Sealed byte representation of The RevokePrescriptionResult-object.*
	RevokePrescriptionResult	*status*	*StatusType*	*Contains the StatusType-object*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*

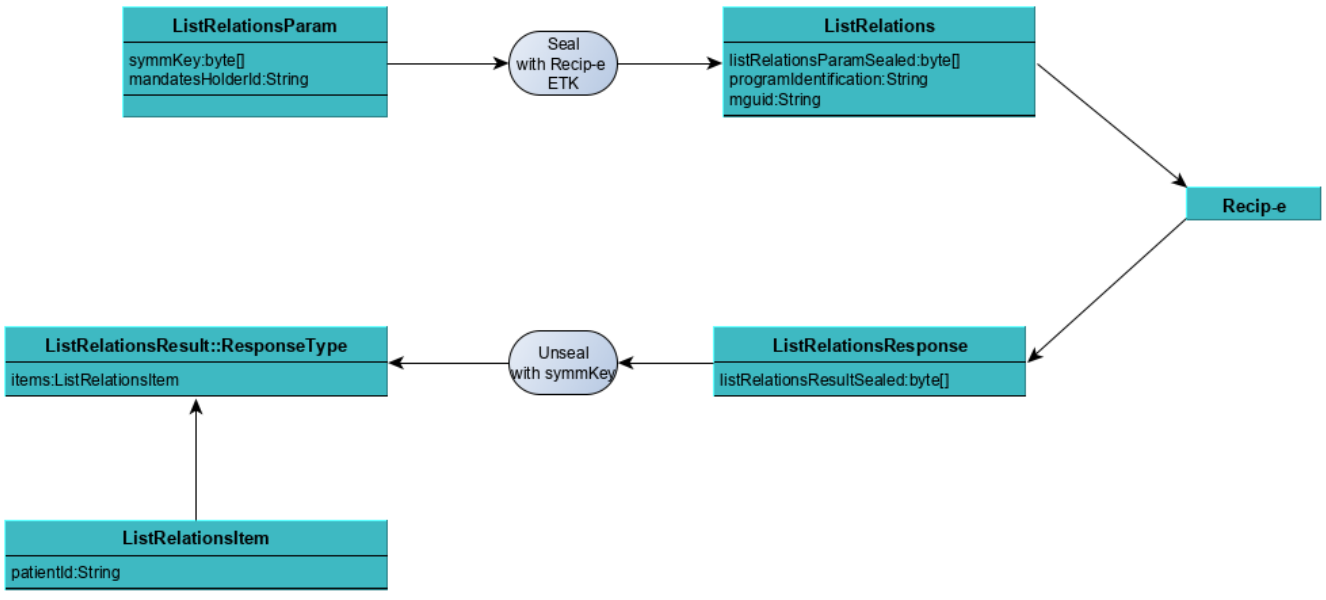


14. LISTRELATIONS

Operation Name	***ListRelations(ListRelations)***
Class Name	***ExecutorIntegrationModuleV4***
Description	*This operation returns the list of patients who have given a mandate to the mandateHolder which is specified in the request.*

Errors	*Specific exceptiondetails are returned in the encrypted part of the response. See StatusType.message. Technical Services Exception such as a Connection failure to the Recip-e server are caught by eHealth and translated into a SOA-fault.*			
Inputs Integration Module	**Request-entity**	**Attribute**	**Type**	**Description**
	ListRelationsParam	*mandateHolderId*	*String*	*The identifier of the mandateholder*
		breakTheGlass	*breakTheGlass*	*breakTheGlass type is defined in UnsealedExecutorServicesV4_schema1*
		symmKey	*Byte []*	*The symmetric key to be used by Recip-e to encrypt the response*
		page(optional)	*Page*	*The page to return. (by default: the last previous 12 months).*
		session(optional)	*Session*	*The session is returned by the server and is contained in the first paging response. This object facilitates faster processing on the server of subsequent listrequests by storing operation specific state.*
	Page	*Month(unused)*	*int*	*The month of interest*
		year(unused)	*int*	*The year of interest*
		pageNumber	*int*	*The pagination number*
	Session		*byte []*	*The session context. This object is unreadable for the client.*
	ListRelations	*ListRelationsParamSealed*	*Byte[]*	*The ListRelationsParam-object sealed with the Recip-e ETK*
		programIdentification	*String*	*The identifier of the calling executor program*
		mguid	*String*	*The mguid to identify the request*
Output Integration Module	**Response-entity**	**Attribute**	**Type**	**Description**
	ListRelationsResponse	*ListRelationsResultSealed*	*Byte []*	*Sealed byte representation of the ListRelationsResult-object. Unseal with the symmetric key from the request.*
	ListRelationsResult	*status*	*StatusType*	*Contains the StatusType-object*

		items	List	*The list of ListRelationsItem-objects*
		hasMoreResults	Boolean	*Flag indicating if there are more results*
		session	Session	*The session is returned by the server and is contained in the first paging response. This object facilitates faster*
	ListRelationsItem	*patientId*	*String*	*The patient who gave the mandateHolderId a mandate to request a (list of) prescription(s) on their behalf.*
		Firstname	*String*	*The firstname of the patient who gave the mandate*
		Lastname	*String*	*The lastname of the patient who gave the mandate*
	StatusType	*statusCode*	*String*	*100: ok* *300: error occurred*
		code	*String*	*The technical code of the error message e.g. error.validation.x.y.z*
		message	*String*	*The error message*
		statusUpdater	*String*	*The last statusupdater in case an operation is not permitted*



Overview of changes of this document

Version Number	Date	Author	Changes (content)	Available as of
Version 1	2019/09/30	BCA	Initial version	
Version 2	2019/11/14	BCA	• Add *activeResults* to *ListRidsHistoryParam* • The *ListReservationsResult contains a list of ListReservationsResultItem-objects in stead of Reservation-objects* • The *ListReservationsResultItem contains the default information of the reservation:* ○ Rid ○ executorId ○ creationdate of the reservation ○ last updatedate of the reservation *And is extended with the following items:* ○ Prescription without proof ○ Creationdate prescription ○ Expirationdate prescription ○ Status of the prescription ○ patientId ○ prescriberId	*Release 2020Q1*
Version 3	2019/12/19	BCA	Correction to • ListOpenPrescriptions • ListRelations	*Release 2020Q1*
Version 4	2020/10/21	BCA	Add firstname and lastname to ListRelationItem	*Release 2020Q3*
Version 5	2021/08/12	NDH	Add contact details and preference to createReservation	
Version 6	2021/12/09	NDH	Add breakTheGlass to ListRelations, ListOpenPrescriptions and ListRidsHistory	