

# **Сервер Модуля Узгодження**

Інструкція користувача

Київ, 2019

# ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	11
<b>1. КОНЦЕПЦІЯ ІНФОРМАЦІЙНОЇ ВЗАЄМОДІЇ .....</b>	<b>12</b>
1.1. Загальні положення .....	12
1.2. Функції модуля узгодження.....	13
<b>2. КЕРІВНИЦТВО ПО ВСТАНОВЛЕННЮ ТА НАЛАГОДЖЕННЮ .....</b>	<b>14</b>
2.1. Інструкція по встановленню служби Сервера Модуля Узгодження .....	14
2.2. Налаштування Служби Сервера Модуля Узгодження.....	15
2.3. Опис установки Клієнта Сервера Модуля Узгодження.....	17
<b>3. ЗАГАЛЬНИЙ ОПИС БІБЛІОТЕКИ .....</b>	<b>18</b>
<b>4. ДИНАМІЧНО ЛІНКУЄМА БІБЛІОТЕКА TMSOFT.GOHUB.CLIENT.DLL .....</b>	<b>20</b>
4.1. Короткий опис бібліотеки .....	20
4.2. Типи даних.....	20
GohubBool .....	20
GohubWChar.....	21
GohubConnection .....	21
GohubDocument.....	21
GohubAttachment .....	21
GohubEData .....	21
GohubPiPackage .....	22
GohubPiPackageToEData .....	22
GohubFdu92.....	22
GohubGu46 .....	22
GohubGu45 .....	22
GohubInformServicesDoc .....	23
GohubDispatchInfo .....	23
4.3. Статуси документів .....	23
4.3.1. Стани перевізного документа .....	23
4.3.2. Стани накопичувальної картки ФДУ-92.....	24
4.3.3. Стани відомостей про користування вагонами / контейнерами ГУ-46.....	24
4.3.4. Стани пам'ятки про подачу / прибирання вагонів і видачу / прийом контейнерів ГУ-45 .....	25
4.4. Основні функції.....	25
4.4.1. Робота з кодовими сторінками .....	25
gohub_set_codepage.....	25
gohub_encoding_codepage.....	25
gohub_encoding_codepage_w .....	26
4.4.2. Підключення до Модуля Узгодження.....	27
gohub_connect .....	27
gohub_connect_w.....	27
gohub_disconnect.....	28
4.4.3. Робота з документами .....	28
gohub_load_document.....	29
gohub_load_document_w .....	29

gohub_create_document.....	30
gohub_create_document_w.....	30
gohub_query_document .....	30
gohub_query_document_w.....	31
gohub_query_next_document .....	31
gohub_document_id.....	31
gohub_document_id_w .....	32
gohub_document_revision .....	32
gohub_document_text .....	32
gohub_document_text_w.....	33
gohub_document_data_text .....	33
gohub_document_data_text_w .....	33
gohub_document_status.....	34
gohub_document_size.....	34
gohub_document_measure_equip_num .....	34
gohub_document_measure_equip_num_w.....	35
gohub_document_set_measure_equip_num .....	35
gohub_document_set_measure_equip_num_w.....	35
gohub_document_get_verified_empty_weight_for_wagon.....	36
gohub_document_set_verified_empty_weight_for_wagon .....	36
gohub_document_warrant_type.....	36
gohub_document_set_warrant_type .....	37
gohub_document_business_unit_num .....	37
gohub_document_business_unit_num_w .....	37
gohub_document_set_business_unit_num .....	37
gohub_document_set_business_unit_num_w.....	38
gohub_document_get_foreign_not_accept .....	38
gohub_send_document .....	38
gohub_save_document.....	39
gohub_save_document_w.....	39
gohub_save_document_data .....	40
gohub_save_document_data_w.....	40
gohub_close_document.....	41
gohub_reclaim_document.....	41
gohub_reclaim_document_w .....	42
gohub_delete_document.....	42
gohub_delete_document_w .....	42
gohub_send_received_document .....	43
gohub_send_received_document_w.....	43
gohub_document_get_otpr.....	44
gohub_document_get_otpr_w .....	44
gohub_document_get_otpr_string.....	44
gohub_document_get_otpr_string_w .....	45
gohub_document_warning.....	45
gohub_document_warning_w .....	45
4.4.4.     Робота зі супровідними документами .....	46
gohub_load_attachment.....	47
gohub_load_attachment_with_user_data .....	47
gohub_load_attachment_w.....	48
gohub_load_attachment_with_user_data_w .....	49
gohub_load_smgs_attachment .....	50
gohub_load_smgs_attachment_with_user_data .....	51
gohub_load_smgs_attachment_w.....	52

gohub_load_smgs_attachment_with_user_data_w .....	52
gohub_send_attachment.....	53
gohub_query_attachment .....	54
gohub_query_attachment_w .....	54
gohub_query_attachment_with_user_data .....	55
gohub_query_attachment_with_user_data_w .....	55
gohub_save_attachment .....	55
gohub_save_attachment_with_user_data .....	56
gohub_save_attachment_w .....	56
gohub_save_attachment_with_user_data_w .....	56
gohub_delete_attachment .....	57
gohub_delete_attachment_w .....	57
gohub_close_attachment .....	57
gohub_attachment_id.....	58
gohub_attachment_id_w.....	58
gohub_attachment_type_code.....	58
gohub_attachment_type_code_w.....	59
gohub_attachment_name .....	59
gohub_attachment_name_w.....	59
gohub_attachment_reg_number.....	59
gohub_attachment_reg_number_w.....	60
gohub_attachment_reg_date .....	60
gohub_attachment_reg_date_w .....	60
gohub_attachment_valid_from .....	61
gohub_attachment_valid_from_w .....	61
gohub_attachment_valid_to.....	61
gohub_attachment_valid_to_w.....	61
gohub_attachment_description .....	62
gohub_attachment_description_w.....	62
gohub_attachment_count .....	62
gohub_attachment_id_by_index.....	63
gohub_attachment_id_by_index_w .....	63
4.4.5. Робота з електронними даними і пакетами попереднього інформування (ПІ) .....	63
gohub_load_edata .....	65
gohub_load_edata_w .....	65
gohub_load_edata_simple.....	66
gohub_load_edata_simple_w.....	67
gohub_send_edata .....	67
gohub_update_edata.....	68
gohub_query_edata.....	68
gohub_query_edata_w.....	68
gohub_query_edata_for_attachment .....	69
gohub_query_edata_for_attachment_w.....	69
gohub_query_next_edata.....	69
gohub_save_edata .....	70
gohub_save_edata_w .....	70
gohub_edata_load_data .....	71
gohub_edata_load_data_w .....	71
gohub_close_edata .....	71
gohub_edata_id .....	72
gohub_edata_id_w .....	72
gohub_edata_revision .....	72
gohub_edata_revision_date .....	72

gohub_edata_revision_date_w .....	73
gohub_edata_doc_type .....	73
gohub_edata_status .....	73
gohub_edata_version .....	73
gohub_edata_version_w.....	74
gohub_edata_attachment_id .....	74
gohub_edata_attachment_id_w.....	74
gohub_query_pi_package .....	75
gohub_query_pi_package_w .....	75
gohub_query_next_pi_package.....	75
gohub_save_pi_package .....	76
gohub_save_pi_package_w .....	76
gohub_close_pi_package .....	76
gohub_pi_package_id .....	77
gohub_pi_package_id_w .....	77
gohub_pi_package_revision .....	77
gohub_pi_package_revision_date .....	78
gohub_pi_package_revision_date_w .....	78
gohub_pi_package_status .....	78
gohub_pi_package_consignment_id .....	78
gohub_pi_package_consignment_id_w .....	79
gohub_pi_package_pipacktoed_count.....	79
gohub_pi_package_pipacktoed .....	79
gohub_pipacktoed_id .....	80
gohub_pipacktoed_id_w.....	80
gohub_pipacktoed_edata_id .....	80
gohub_pipacktoed_edata_id_w .....	80
gohub_pipacktoed_pi_package_id .....	81
gohub_pipacktoed_pi_package_id_w .....	81
gohub_pipacktoed_note .....	81
gohub_pipacktoed_note_w .....	82
gohub_pipacktoed_edata_version .....	82
gohub_pipacktoed_edata_version_w.....	82
gohub_add_edata_to_pi_package .....	82
gohub_add_edata_to_pi_package_w.....	83
4.4.6. Робота з накопичувальними картками ФДУ-92 .....	83
gohub_query_fdu92.....	84
gohub_query_fdu92_w.....	84
gohub_query_fdu92_by_number .....	85
gohub_query_fdu92_by_number_w .....	85
gohub_create_fdu92.....	86
gohub_create_fdu92_w .....	86
gohub_load_fdu92.....	86
gohub_load_fdu92_w .....	87
gohub_send_fdu92 .....	87
gohub_query_next_fdu92 .....	87
gohub_fdu92_id.....	88
gohub_fdu92_id_w .....	88
gohub_fdu92_status .....	88
gohub_fdu92_revision .....	89
gohub_fdu92_text.....	89
gohub_fdu92_text_w.....	89
gohub_fdu92_size.....	89

gohub_fdu92_signer_info .....	90
gohub_fdu92_signer_info_w .....	90
gohub_fdu92_sign_time .....	90
gohub_fdu92_sign_time_w .....	91
gohub_fdu92_sign_name_w.....	91
gohub_fdu92_has_signature .....	91
gohub_save_fdu92.....	91
gohub_save_fdu92_w.....	92
gohub_reject_fdu92.....	92
gohub_reject_fdu92_w.....	93
gohub_close_fdu92.....	93
4.4.7. Робота з відомостями про користування вагонами / контейнерами ГУ-46 .....	93
gohub_query_gu46 .....	94
gohub_query_gu46_w .....	95
gohub_query_next_gu46.....	95
gohub_create_gu46 .....	96
gohub_create_gu46_w .....	96
gohub_load_gu46 .....	96
gohub_load_gu46_w.....	97
gohub_send_gu46.....	97
gohub_gu46_id .....	97
gohub_gu46_id_w.....	98
gohub_gu46_status.....	98
gohub_gu46_revision.....	98
gohub_gu46_text.....	99
gohub_gu46_text_w .....	99
gohub_gu46_size .....	99
gohub_gu46_signer_info .....	100
gohub_gu46_signer_info_w .....	100
gohub_gu46_sign_time.....	100
gohub_gu46_sign_time_w.....	100
gohub_gu46_sign_name_w.....	101
gohub_gu46_has_signature.....	101
gohub_save_gu46 .....	101
gohub_save_gu46_w .....	102
gohub_reject_gu46 .....	102
gohub_reject_gu46_w .....	103
gohub_close_gu46 .....	103
4.4.8. Робота з пам'ятками про подачу / прибирання вагонів та видачу / прийом контейнерів ГУ-45.....	103
gohub_query_gu45 .....	104
gohub_query_gu45_w .....	104
gohub_query_next_gu45.....	105
gohub_gu45_id .....	105
gohub_gu45_id_w.....	106
gohub_gu45_status.....	106
gohub_gu45_revision.....	106
gohub_gu45_text.....	107
gohub_gu45_text_w .....	107
gohub_gu45_size .....	107
gohub_gu45_signer_info .....	107
gohub_gu45_signer_info_w .....	108
gohub_gu45_sign_time.....	108
gohub_gu45_sign_time_w.....	108

gohub_gu45_sign_name_w .....	109
gohub_gu45_has_signature.....	109
gohub_save_gu45 .....	109
gohub_save_gu45_w .....	110
gohub_close_gu45 .....	110
4.4.9. Отримання друкованої форми документ .....	110
gohub_query_and_save_document_printable_form .....	111
gohub_query_and_save_document_printable_form_w .....	111
4.4.10. Робота з фільтрами для запитуваних документів.....	112
gohub_clear_all_filters .....	113
gohub_set_filter_by_document_status.....	113
gohub_set_filter_by_document_number.....	113
gohub_set_filter_by_document_number_w.....	114
gohub_set_filter_by_wagon_number .....	114
gohub_set_filter_by_wagon_number_w.....	114
gohub_set_filter_by_departure_client.....	115
gohub_set_filter_by_departure_client_w .....	115
gohub_set_filter_by_departure_payer.....	116
gohub_set_filter_by_departure_payer_w.....	116
gohub_set_filter_by_departure_station.....	116
gohub_set_filter_by_departure_station_w.....	117
gohub_set_filter_by_arrival_client.....	117
gohub_set_filter_by_arrival_client_w .....	117
gohub_set_filter_by_arrival_payer.....	118
gohub_set_filter_by_arrival_payer_w.....	118
gohub_set_filter_by_arrival_station.....	119
gohub_set_filter_by_arrival_station_w.....	119
gohub_get_filter_by_document_status .....	119
gohub_get_filter_by_document_number .....	120
gohub_get_filter_by_document_number_w .....	120
gohub_get_filter_by_wagon_number .....	120
gohub_get_filter_by_wagon_number_w .....	121
gohub_get_filter_by_departure_client .....	121
gohub_get_filter_by_departure_client_w.....	121
gohub_get_filter_by_departure_payer .....	122
gohub_get_filter_by_departure_payer_w .....	122
gohub_get_filter_by_departure_station .....	122
gohub_get_filter_by_departure_station_w .....	123
gohub_get_filter_by_arrival_client.....	123
gohub_get_filter_by_arrival_client_w.....	123
gohub_get_filter_by_arrival_payer .....	124
gohub_get_filter_by_arrival_payer_w .....	124
gohub_get_filter_by_arrival_station .....	124
gohub_get_filter_by_arrival_station_w .....	125
4.4.11. Перевірка електронно-цифрового підпису .....	125
gohub_document_has_signature .....	125
gohub_document_check_signature.....	125
gohub_document_signature_name.....	126
gohub_document_signature_name_w.....	126
gohub_document_signer_info.....	126
gohub_document_signer_info_w .....	127
gohub_document_sign_time .....	127
gohub_document_sign_time_w .....	127

4.4.12.	Накладення електронно-цифрового підпису.....	128
	gohub_open_private_key .....	128
	gohub_open_private_key_w .....	128
	gohub_open_private_key_by_bytes.....	129
	gohub_open_private_key_by_bytes_w.....	129
	gohub_open_private_key_from_path .....	130
	gohub_open_private_key_from_path_w .....	130
	gohub_private_key_owner_name.....	131
	gohub_private_key_owner_name_w .....	131
	gohub_private_key_owner_info.....	131
	gohub_private_key_owner_info_w .....	132
	gohub_sign_document .....	132
	gohub_close_private_key .....	132
4.4.13.	Операції з файлами електронних ключів .....	133
	gohub_mount_file_key .....	133
	gohub_mount_file_key_w .....	134
	gohub_unmount_file_key.....	134
	gohub_unmount_file_key_w .....	134
	gohub_query_mounted_file_keys .....	134
	gohub_mounted_file_key_id .....	135
	gohub_mounted_file_key_id_w .....	135
	gohub_mounted_file_key_dir.....	135
	gohub_mounted_file_key_dir_w.....	136
4.4.14.	Робота з АС «Месплан».....	136
	gohub_get_mp_months.....	136
	gohub_get_mp_months_w.....	137
	gohub_query_and_save_orders_for_month .....	137
	gohub_query_and_save_orders_for_month_w .....	137
	gohub_query_and_save_orders_for_month_with_relogin.....	138
	gohub_query_and_save_orders_for_month_with_relogin_w .....	139
4.4.15.	Робота з документами інформаційних послуг.....	139
	gohub_query_inform_services_document .....	140
	gohub_query_next_inform_services_document .....	140
	gohub_save_inform_services_document .....	140
	gohub_save_inform_services_document_w .....	141
	gohub_saveXml_inform_services_document .....	141
	gohub_saveXml_inform_services_document_w .....	142
	gohub_close_inform_services_document.....	142
	gohub_inform_services_document_id .....	142
	gohub_inform_services_document_revision .....	143
	gohub_inform_services_document_filename .....	143
	gohub_inform_services_document_filename_w .....	143
	gohub_inform_services_document_comment.....	143
	gohub_inform_services_document_comment_w .....	144
	gohub_inform_services_document_created_date.....	144
	gohub_inform_services_document_created_date_w .....	144
	gohub_inform_services_document_doc_date .....	145
	gohub_inform_services_document_doc_date_w .....	145
4.4.16.	Обробка помилок.....	145
	gohub_last_error_code.....	145
	gohub_last_error_title .....	145
	gohub_last_error_title_w .....	146
	gohub_last_error_text .....	146



gohub_last_error_text_w .....	146
4.4.17. Робота з інформацією про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів (за останні 5 днів від поточної дати) .....	146
gohub_query_dispatch_info .....	146
gohub_query_dispatch_info_w .....	147
gohub_document_info_description .....	147
gohub_document_info_description_w .....	147
gohub_document_info_count .....	148
gohub_document_info_is_empty .....	148
gohub_document_info_is_empty_w .....	148
gohub_document_info_wag_owner .....	149
gohub_document_info_wag_owner_w .....	149
gohub_document_info_date .....	149
gohub_document_info_date_w .....	150
gohub_document_info_number .....	150
gohub_document_info_number_w .....	151
gohub_document_info_type .....	151
gohub_document_info_type_w .....	151
gohub_close_dispatch_info .....	152
<b>4.5. Коди помилок .....</b>	<b>153</b>
<b>4.6. Приклади використання .....</b>	<b>155</b>
а) Вивід на екран інформації про помилку .....	155
б) Завантаження документа з файлу та передача в АС «Клієнт УЗ» .....	155
в) Запит документів з АС «Клієнт УЗ» .....	157
г) Інші приклади використання .....	158
<b>5. .NET БІБЛІОТЕКА - TMSOFT.GOHUB.CLIENT.NET.DLL .....</b>	<b>161</b>
5.1. Перелік типів .....	161
5.2. Стани перевізного документа .....	161
5.3. GohubConnection .....	161
5.4. GohubDocument .....	164
5.5. GohubAttachment .....	166
5.6. GohubEDData .....	167
5.7. GohubPiPackage .....	167
5.8. GohubPiPackageToEDData .....	168
5.9. GohubFdu92 .....	168
5.10. GohubGu46 .....	169
5.11. GohubGu45 .....	169
5.12. GohubDocumentFilter .....	169
5.13. GohubSigner .....	170
5.14. GohubClient .....	171
5.15. GohubException .....	171
5.16. GohubErrCode .....	171
5.17. GohubInformServicesDoc .....	173
5.18. GohubDispatchInfo .....	173
5.19. Приклади використання .....	173
а) Завантаження документа з файлу та передача в АС «Клієнт УЗ» .....	173
б) Запит документів з АС «Клієнт УЗ» .....	174
в) Інші приклади використання .....	175
<b>6. COM/OLE БІБЛІОТЕКА .....</b>	<b>177</b>
6.1. Стани перевізного документа .....	177

6.2.	Ідентифікатори інтерфейсів .....	177
6.3.	Інтерфейс IGohubClient.....	178
6.4.	Інтерфейс IGohubDocument .....	179
6.5.	Інтерфейс IGohubAttachment .....	180
6.6.	Інтерфейс IGohubEData .....	180
6.7.	Інтерфейс IGohubPiPackage .....	181
6.8.	Інтерфейс IGohubPiPackageToEData .....	181
6.9.	Інтерфейс IGohubConnection .....	181
6.10.	Інтерфейс IGohubSignerInfo.....	183
6.11.	Інтерфейс IGohubError .....	183
6.12.	Інтерфейс IGohubFdu92 .....	183
6.13.	Інтерфейс IGohubGu46 .....	184
6.14.	Інтерфейс IGohubGu45 .....	184
6.15.	Інтерфейс IGohubInformServicesDocument .....	185
6.16.	Інтерфейс IDispatchInfo .....	185
6.17.	Приклади використання .....	186
а)	Вивід на екран інформації про помилку .....	186
б)	Завантаження документа з файлу та передача в АС «Клієнт УЗ» .....	186
в)	Запит документів з АС «Клієнт УЗ».....	187
г)	Інші приклади використання.....	188

<b>ДОДАТОК А. ФАЙЛ <i>GOHUB.CLIENT.H</i> .....</b>	<b>190</b>
--	------------

<b>ДОДАТОК Б. ФАЙЛ <i>GOHUB.CLIENT.ERRORS.H</i> .....</b>	<b>204</b>
---	------------

<b>ДОДАТОК В. ФАЙЛ ОПИСУ СОМ-ІНТЕРФЕЙСІВ <i>GOHUBCLIENTCOM.IDL</i> .....</b>	<b>207</b>
--	------------

## Перелік скорочень

АС	–	автоматизована система
СМУ	–	Сервер Модуля Узгодження
АРМ ВВ	–	автоматизоване робоче місце вантажовідправника
ЕПД	–	електронний перевізний документ
ЕД	–	електронні дані документа
ЕЦП	–	електронно-цифровий підпис
ЦСК	–	центр сертифікації ключів
XML	–	extensible markup language
DLL	–	динамічно лінкуєма бібліотека
УЗ	–	Українські залізниці
ПІ	–	попереднє інформування

# 1. Концепція інформаційної взаємодії

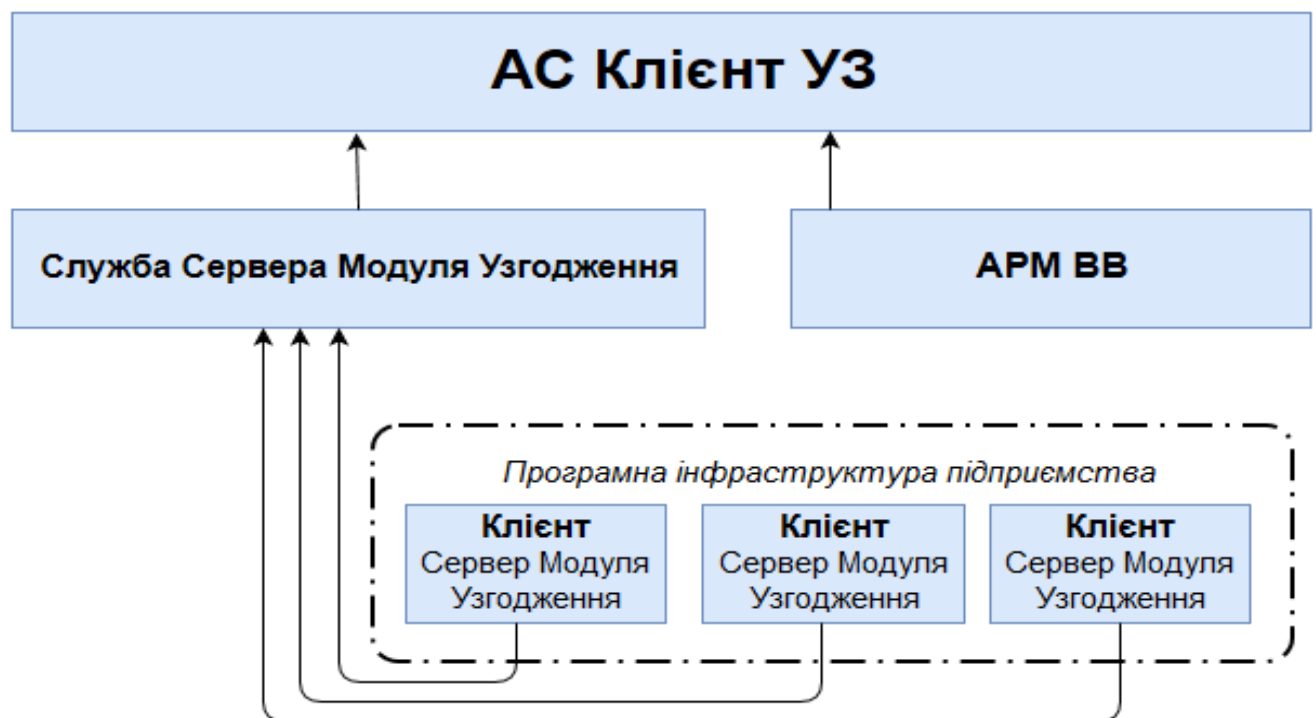
## 1.1. Загальні положення

*Сервер Модуля Узгодження* – модуль дозволяє прикладного програмного забезпечення взаємодіяти з АС «Клієнт УЗ». Користувачі отримують можливість формувати перевізні документи за допомогою власних програм з подальшою передачею їх в АС «Клієнт УЗ». Крім того, *Сервер Модуля Узгодження* дозволяє призначенням для користувача програмам синхронізувати стан документів (ЕПД) електронних даних (ЕД) і чернеток користувальницької програмної системи зі станом відповідних проєктів системи АС «Клієнт УЗ».

Фізично *Сервер Модуля Узгодження* є службою Windows, запущену на одному ПК, до якої підключаються *Клієнти СМУ*.

Для полегшення реалізації взаємодії прикладних програм користувачів з *Клієнтом СМУ* до складу поставки включено кілька варіантів бібліотек *Клієнта*, надають прикладним програмам дружній програмний інтерфейс для наступних мов програмування: С, С ++, COM / OLE, С #, ін. мови платформи. *NET*.

На малюнку 1 представлена приблизна схема взаємодії прикладної програми користувача з АС «Клієнт УЗ» з використанням *Сервера Модуля Узгодження*.



Малюнок 1

## **1.2. Функції модуля узгодження**

- передача перевізних і супровідних документів, створених в призначених для користувача програмах, в АС «Клієнт УЗ»;
- отримання інформації про зміни (і списку змін), що відбулися в ЕПД, ЕД, чернетках і супровідних документах з конкретного моменту;
- отримання окремих ЕПД, ЕД, чернеток і супровідних документів за ідентифікатором з АС «Клієнт УЗ»;
- передача дозаповнення в призначених для користувача програмах ЕПД та ЕД по прибуттю в АС «Клієнт УЗ»;
- виконання накладення ЕЦП;
- виконання перевірки ЕЦП;
- відкликання документів та супровідних документів;
- видалення чернеток та супровідних документів;
- підключення файлів електронних ключів і управління підключеними ключами
- отримання заявок з АС «Месплан»

## **2. Керівництво по встановленню та налагодженню**

### **2.1. Інструкція по встановленню служби Сервера Модуля Узгодження**

Є два варіанти установки Служби сервера Модуля Узгодження:

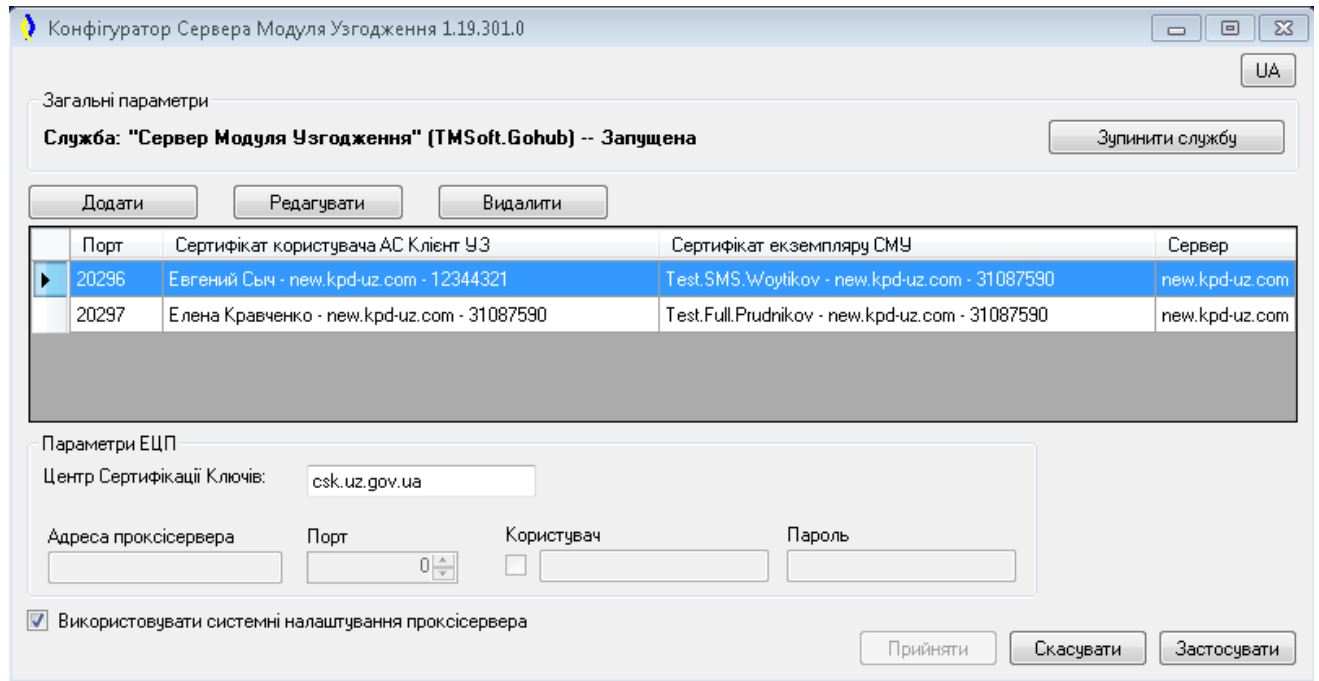
- потрібно запустити інсталятор *tmsoft.gohub.service.setup.msi* (якщо у Вас вже встановлено *.NET Framework 3.5 Servis Pack 1*);
- потрібно запустити інсталятор *tmsoft.gohub.service.setup.exe* (якщо потрібна повна Встановлення, включаючи *.NET Framework*)

В процесі інсталяції буде запропоновано вибрати директорію установки, за замовчуванням - *C: \ Program Files \ TMSoft \ Сервер Модуля Узгодження*.

В результаті установки Служба СМС буде інстальована в систему як служба Windows з ім'ям *TMSoft.Gohub* (в переліку служб відображається з ім'ям «Модуль Узгодження»). Засобами операційної системи можна вибрати, чи запускати служби (ручний або автоматичний, за замовчуванням - автоматичний) і стартувати службу.

## 2.2. Налаштування Служби Сервера Модуля Узгодження

Для налаштування Служби СМУ в комплект поставки включена утиліта *Конфігуратор Сервера Модуля Узгодження*, яка запускається після завершення інсталяції, а також доступна в основі події, в системне меню: *Пуск -> TMSoft -> Сервер Модуль Узгодження -> Конфігуратор*.



Вікно: *Конфігуратор Сервера Модуля Узгодження*

*Конфігуратор* дозволяє налаштувати наступні властивості:

- Запуск Служби СМУ: кнопка «Запустити службу» / «Зупинити службу» запускає / зупиняє Службу СМУ;
- Створювати, редагувати і видаляти дані про підключення Сервера Модуля Узгодження до АС «Клієнт» УЗ;
- Установки для перегляду ЕЦП:
  - Адреса сервера Центру Сертифікації Ключів (по замовчуванням - «*csk.uz.gov.ua*»);
  - Налаштування проксісервера - адреса і порт;
  - Налаштування користувача локальної мережі - ім'я та пароль.

Додати \ Редагувати

Параметри взаємодії з АС "Клієнт УЗ"

Номер TCP-порту:

Сертифікат екземпляра:

Сервер:

Сертифікат користувача:

Пароль:

В

ікно  
додав  
ання  
/  
редаг  
уван  
ня  
дани  
х про  
підкл  
ючен  
ня

- Номер TCP-порту, по якому здійснюється взаємодія *Служби* і *Клієнта СМС*);
- Параметри взаємодії з АС «Клієнт УЗ»:
  - Сертифікат примірника - сертифікат надає можливість роботи Службі СМС і взаємодіяти з АС «Клієнт УЗ»;
  - Кнопка «Імпорт ...» надає можливість внести дані про сертифікат примірника з файлу сертифіката;
  - В поле «Сервер» задається адреса сервера АС «Клієнт УЗ», який автоматично встановлюється з сертифіката користувача АС «Клієнт УЗ»;
  - В поле «Ім'я користувача» вибирається користувач АС «Клієнт УЗ», який встановлюється з сертифіката користувача АС «Клієнт УЗ»;
  - В поле «Пароль користувача» задається пароль доступу до АС «Клієнт УЗ»;
  - Кнопка «Сертифікати ...» відкриває браузер дозволів, які стосуються користувачам Служби СМС, від яких йде звернення до АС «Клієнт УЗ» і зберігаються в операційній системі;
  - Кнопка «Імпорт сертифікатів користувачів ...» надає можливість внести дані про користувача, від якого йде звернення до АС «Клієнт УЗ» з файлу сертифіката;



## 2.3. Опис установки *Клієнта Сервера Модуля Узгодження*

Є два варіанти установки Клієнта сервера Модуля Узгодження:

- потрібно запустити інсталятор *tmsoft.gohub.client.setup.msi* (якщо у Вас вже встановлено *.NET Framework 2.0*);
- потрібно запустити інсталятор *tmsoft.gohub.client.setup.exe* (якщо потрібна повна Встановлення, включаючи *.NET Framework*).

В процесі інсталяції буде запропоновано вибрати директорію установки, за замовчуванням - *C: \ Program Files \ TMSoft \ Клієнт Модуля Узгодження*.

В результаті установки *Клієнт СМС* в теці установки розміщується наступні файли:

- Тека ***bin*** містить підготовлений до запуску файл консолі (його можна побачити і в розділі «Пуск / Всі програми / ТМСофт / Клієнт модуля узгодження» як «Консоль Модуля Узгодження») і три файли різних варіантів бібліотеки Клієнта:
  - ***TMSoft.gohub.client.dll*** - динамічно лінкуємая бібліотека (DLL) для сімейства операційних систем Windows, починаючи з Windows 2000 і вище;
  - ***TMSoft.gohub.client.com.dll*** - COM / OLE про'єкт;
  - ***TMSoft.Gohub.Client.Net.dll*** - .NET версія бібліотеки.

*Примітка: бібліотеки можна вільно переміщувати, копіювати і т.д., крім бібліотеки COM / OLE, яка при установці реєструється в системі із зазначенням шляху розміщення файлу.*

- Тека ***doc*** містить два файли - Керівництво користувача і *Changes.txt* з описами змін в попередніх версіях;
- Тека ***inc*** містить файли для лінування бібліотек - заголовки *gohub.client.h* і *gohub.client.errors.h*, для динамічно лінкуємой бібліотеки і файл *gohubclientcom.idl* з оголошенням інтерфейсів для COM / OLE версії бібліотеки;
- Тека ***lib*** містить два файли: статична бібліотека *TMSoft.gohub.client.lib* і бібліотека типів *TMSoft.gohub.client.com.tlb* для додатків, що використовують COM / OLE версію.
- Тека ***samples*** містить xml-файли зразків повідомлень для формування електронних перевізних документів різних видів відправлень, а так же файл *Gohub.Client.Test.xls*, з макросами-прикладми використання COM / OLE версії бібліотеки;
- Тека ***schemas*** містить файли *xml-Схема* (. Xsd) з формалізованим описом структур xml-повідомлень електронних перевізних документів.

### 3. Загальний опис бібліотеки

Прикладна бібліотека «Клієнт Сервера Модуля Узгодження» (далі - *Бібліотека*) призначена для спрощення взаємодії прикладних програм із *Службою Сервера Модуля Узгодження*. *Бібліотека* реалізована відповідно до вимог, викладених в «Концепції інформаційної взаємодії» *Сервера Модуля Узгодження*.

Фізично *Бібліотека* складається з декількох варіантів бібліотеки *TMSoft.gohub.client*, під різними мовами програмування:

- *TMSoft.gohub.client.dll* - динамічно лінуємою бібліотекою (DLL) для сімейства операційних систем Windows, починаючи з Windows 2000 і вище;
- *TMSoft.gohub.client.com.dll* - COM / OLE про'єкт;
- *TMSoft.Gohub.Client.Net.dll* - .NET версія бібліотеки.

*Бібліотека* надає набір простих і в той же час ефективних засобів діагностики помилок, що виникають при її використанні.

Програмний інтерфейс *Бібліотеки* надає прикладним програмам можливість вирішувати такі завдання:

- Відправляти в АС «Клієнт УЗ» перевізні документи, створені поза системою АС «Клієнт УЗ».
- Відслідковувати появу в АС «Клієнт УЗ» нових документів і модифікацію існуючих.
- Запитувати з АС «Клієнт УЗ» окремі документи по їх унікальним ідентифікаторів.
- Виконувати перекодування документів і отримувати детальну інформацію про можливі помилки, що виникають при роботі з *Бібліотекою*.
- Відкликати документи і видаляти чернетки.
- Накладати ЕЦП на перевізні документи.
- Виробляти перевірку ЕЦП, накладеної на перевізні документи.
- Отримувати заявки з АС «Месплан»

Для взаємодії з АС «Клієнт УЗ» прикладна програма повинна підключитися до *Сервера Модуля Узгодження* за допомогою функції «Створити\_з'єднання». Результатом виклику цієї функції є дескриптор (показчик) про'єкта або про'єкт з'єднання, який буде використовуватися у всіх операціях взаємодії з АС «Клієнт УЗ»: відправлення та запиту перевізних документів, а також запиту списків модифікацій.

Для відправлення чернетки перевізного документа в АС «Клієнт УЗ» прикладна програма спочатку повинна створити про'єкт документа за допомогою функції «Створити\_документ» , або завантажити документ з файлу за допомогою функції «Завантажити\_документ». Після цього можна відправити документ в АС «Клієнт УЗ» за

допомогою функції «Відправити\_документ». Формат чернеток перевізних документів, що відправляються в АС «Клієнт УЗ», повинен відповідати вимогам викладеним в документі «Структура, склад та формат реквізитів та атрибутів електронного перевізного документа», опублікованому на офіційному сайті УЗ.

В результаті відправлення в АС «Клієнт УЗ» документ отримує унікальний ідентифікатор по якому можна в майбутньому запросити документ з АС «Клієнт УЗ» протягом усього його життєвого циклу. Для запиту документа з АС «Клієнт УЗ» служить функція «Запросити\_документ» .

Отримані документи можна зберегти в файл за допомогою функції «Зберегти\_документ» , або отримати текст документа у вигляді рядка за допомогою функції «Текст\_документа» . Ця функція повертає текст документа (без xml-заголовка) в поточній кодуванні, яка може бути задана за допомогою функції «Встановити\_кодування» (за замовчуванням - 1251).

Документи, отримані з АС «Клієнт УЗ» за замовчуванням мають кодування *utf-8* (кодова сторінка 65001). При збереженні документа в файл за допомогою функції «Зберегти\_документ» ви можете вказати кодову сторінку в якій хочете зберегти документ .

Крім унікального ідентифікатора, кожен документ в системі АС «Клієнт УЗ» має таку важливу властивість як номер ревізії. Номер ревізії, так само як і ідентифікатор, призначаються документу при першому попаданні в систему. Однак, на відміну від ідентифікатора, номер ревізії не є постійною властивістю і змінює своє значення при кожній модифікації документа. Всі модифікації по всім документам системи АС «Клієнт УЗ» мають наскрізну нумерацію.

Номери ревізії дозволяють зовнішнім системам відслідковувати зміни в документах, що знаходяться в АС «Клієнт УЗ». За допомогою функції «Завантажити\_наступний\_документ» додатка можуть отримувати документи відповідно до послідовності їх послідовності їх модифікацій.

Всі назви функцій наведені українською мовою спеціально, так як семантика в різних *бібліотеках* відрізняється, але сутність логіки дій і команд не змінюється.

## 4. Динамічно лінкуєма бібліотека `tmsoft.gohub.client.dll`

### 4.1. Короткий опис бібліотеки

*Gohub.client.dll* є динамічно лінкуємою бібліотекою (DLL) для сімейства операційних систем Windows, починаючи з Windows 2000 і вище. Для роботи перших версій *Бібліотеки* будуть потрібні також платформа *.NET Framework* версії не нижче 2.0, а також бібліотеки часу виконання *Microsoft CRT* . Дистрибутиви цих додаткових компонентів входять в комплект поставки *Бібліотеки* .

Додатково в комплект поставки включені: статична бібліотека *gohub.client.lib* для лінковки з *Бібліотекою* програм, написаних мовою *Microsoft Visual C / C ++* , та заголовки *gohub.client.h* , що містить декларації всіх функцій, типів і констант, що становлять програмний інтерфейс *Бібліотеки* .

У *Бібліотеці* є ряд функцій працюють з малими даними виключно в юнікодовому кодуванні UTF-16 Little Indian (далі - *UTF-16*) , всі вони містять в назві букву **W** .

Більш детальний опис програмного інтерфейсу бібліотеки описано в наступних пунктах. Так само можна переглянути заголовки *gohub.client.h* (Додаток А) і *gohub.client.errors.h* (Додаток Б).

Для полегшення і прискорення знайомства з програмним інтерфейсом *Бібліотеки* служать приклади її використання, наведені в п.4.5.

### 4.2. Типи даних

GohubBool	Булева змінна
GohubWChar	Символ представлений в кодуванні UTF-16
GohubConnection	З'єднання з <i>Модулем Узгодження</i>
GohubDocument	Перевізні документи
GohubAttachment	Супровідні документи
GohubEData	Електронні дані попереднього інформування
GohubPiPackage	Пакети попереднього інформування
GohubPiPackageToEData	Інформація про електронні дані в пакеті попереднього інформування
GohubFdu92	Документи ФДУ-92
GohubGu46	Документи ГУ-46
GohubGu45	Документи ГУ-45
GohubInformServicesDoc	Документи інформаційних послуг
GohubDispatchInfo	Перелік № Замовлень для перевізного документу

#### **GohubBool**

Булева змінна.

Об'явлення:

```
typedef int GohubBool;
```

## GohubWChar

Рядкова змінна в кодуванні UTF-16.

Об'явлення:

```
typedef unsigned char GohubWChar;
```

## GohubConnection

Об'єкти цього типу служать для подання підключення прикладної програми до служби *Модуля Узгодження*

Об'явлення:

```
typedef struct GohubConnection GohubConnection;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можлива тільки через відкриті функції *Бібліотеки*, які оперують покажчиками на ці про'єкти .

## GohubDocument

Об'єкти цього типу служать для подання електронних перевізних документів.

Об'явлення:

```
typedef struct GohubDocument GohubDocument;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можливо тільки через відкриті функції *Бібліотеки*, які оперують покажчиками на ці про'єкти .

## GohubAttachment

Об'єкти цього типу служать для подання електронних супровідних документів.

Об'явлення:

```
typedef struct GohubAttachment GohubAttachment;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можливо тільки через відкриті функції *Бібліотеки*, які оперують покажчиками на ці про'єкти .

## GohubEData

Об'єкти цього типу служать для подання електронних даних попереднього інформування.

Об'явлення:

```
typedef struct GohubEData GohubEData;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можливо тільки через відкриті функції *Бібліотеки*, які оперують покажчиками на ці про'єкти .

## **GohubPiPackage**

Об'єкти цього типу служать для подання пакетів попереднього інформування .

Об'явлення:

```
typedef struct GohubPiPackage GohubPiPackage;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можливо тільки через відкриті функції Бібліотеки, які оперують покажчиками на ці про'єкти .

## **GohubPiPackageToEData**

Об'єкти цього типу служать для представлення інформації про електронні дані в пакеті попереднього інформування.

Об'явлення:

```
typedef struct GohubPiPackageToEData GohubPiPackageToEData;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можливо тільки через відкриті функції Бібліотеки, які оперують покажчиками на ці про'єкти .

## **GohubFdu92**

Об'єкти цього типу служать для подання електронних накопичувальних карток ФДУ-92.

Об'явлення:

```
typedef struct GohubFdu92 GohubFdu92;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можливо тільки через відкриті функції Бібліотеки, які оперують покажчиками на ці про'єкти.

## **GohubGu46**

Об'єкти цього типу служать для подання електронних відомостей про користування вагонами / контейнерами ГУ-46 .

Об'явлення:

```
typedef struct GohubGu46 GohubGu46;
```

Примітки:

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з про'єктами цього типу можливо тільки через відкриті функції Бібліотеки, які оперують покажчиками на ці про'єкти.

## **GohubGu45**

Об'єкти цього типу служать для подання електронних пам'яток про подачу / прибирання вагонів і видачу / прийом контейнерів ГУ-45.

Об'явлення:

```
typedef struct GohubGu45 GohubGu45;
```

**Примітки:**

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з проєктами цього типу можливо тільки через відкриті функції Бібліотеки, які оперують покажчиками на ці проєкти.

## ***GohubInformServicesDoc***

Об'єкти цього типу служать для подання документів інформаційних послуг.

**Об'явлення:**

```
typedef struct GohubInformServicesDoc GohubInformServicesDoc;
```

**Примітки:**

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з проєктами цього типу можливо тільки через відкриті функції Бібліотеки, які оперують покажчиками на ці проєкти.

## ***GohubDispatchInfo***

Об'єкти цього типу служать для отримання переліку інформації про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів(за останні 5 днів від поточної дати).

**Об'явлення:**

```
typedef struct GohubDispatchInfo GohubDispatchInfo;
```

**Примітки:**

Повне визначення цього типу недоступно прикладній програмі. Об'єкти цього типу, а також процедури їх створення та знищення вміщені в Бібліотеці. Взаємодія з проєктами цього типу можливо тільки через відкриті функції Бібліотеки, які оперують покажчиками на ці проєкти.

### **4.3. Статуси документів**

#### **4.3.1. Стани перевізного документа**

Коди стану перевізного документа в *Бібліотеці* представлені перерахуванням ( *enum* ) `Gohub DocumentStatus`, що містить набір констант.

*Зведена таблиця кодів станів перевізного документа*

<code>gohub_document_status_unknown = 0</code>	Статус невідомий
<code>gohub_document_status_draft = 1</code>	Чернетка
<code>gohub_document_status_sending = 2</code>	Документ передається товарному касиру
<code>gohub_document_status_registered = 3</code>	Документ переданий товарному касиру
<code>gohub_document_status_reclaiming = 4</code>	Документ відкликається від товарного касира
<code>gohub_document_status_accepted = 5</code>	Вантаж прийнято до перевезення
<code>gohub_document_status_delivered = 6</code>	Вантаж прибув
<code>gohub_document_status_recieved = 7</code>	Вантаж отримано одержувачем
<code>gohub_document_status_uncredited = 8</code>	Документ розкредитовано товарним касиром
<code>gohub_document_status_recieved_draft = 9</code>	Вантаж отримано одержувачем і редагується
<code>gohub_document_status_recieved_sending = 10</code>	Вантаж отримано одержувачем і переданий товарному касиру
<code>gohub_document_status_recieved_reclaiming = 11</code>	Вантаж отримано одержувачем і

	відкликається від товарного касира
gohub_document_status_canceled = 12	Документ зіпсований товарним касиром
gohub_document_status_locked = 13	Документ заблокований

Більш детальний опис програмного інтерфейсу бібліотеки можна отримати з файлу gohub.client.h (Додаток А).

#### 4.3.2. Стани накопичувальної картки ФДУ-92

Коди стану накопичувальної картки ФДУ-92 в *Бібліотеці* представлені перерахуванням (enum) GohubFdu92Status, що містить набір констант.

*Зведена таблиця кодів станів накопичувальної картки ФДУ-92*

gohub_fdu92_status_unknown = -1	Статус невідомий
gohub_fdu92_status_approoving = 0	Документ отримано на узгодження
gohub_fdu92_status_approving_modified = 1	Документ отримано на узгодження і відредагований користувачем
gohub_fdu92_status_confirmed = 2	Документ узгоджений та підписаний
gohub_fdu92_status_canceled = 3	Документ скасований
gohub_fdu92_status_agreed_noted_sending = 4	Документ погоджений з правками, відправляється
gohub_fdu92_status_agreed = 5	Документ погоджений
gohub_fdu92_status_agreed_noted = 6	Документ погоджений з правками
gohub_fdu92_status_expired = 7	Документ не узгоджений вчасно
gohub_fdu92_status_agreed_sending = 8	Документ погоджений, відправляється
gohub_fdu92_status_confirmed_paper = 9	Документ підтверджений товарним касиром
gohub_fdu92_status_paper = 60	Паперовий документ

#### 4.3.3. Стани відомостей про користування вагонами / контейнерами ГУ-46

Коди стану відомостей про користування вагонами / контейнерами ГУ-46 в *Бібліотеці* представлені перерахуванням (enum) GohubGU46Status, що містить набір констант.

*Зведена таблиця кодів станів відомості користування вагонами / контейнерами ГУ-46*

gohub_gu46_status_unknown = -1	Статус невідомий
gohub_gu46_status_approoving = 0	Документ отримано на узгодження
gohub_gu46_status_approving_modified = 1	Документ отримано на узгодження і відредагований користувачем
gohub_gu46_status_confirmed = 2	Документ узгоджений та підписаний
gohub_gu46_status_canceled = 3	Документ скасований
gohub_gu46_status_agreed_noted_sending = 4	Документ погоджений з правками, відправляється
gohub_gu46_status_agreed = 5	Документ погоджений
gohub_gu46_status_agreed_noted = 6	Документ погоджений з правками
gohub_gu46_status_expired = 7	Документ не узгоджений вчасно
gohub_gu46_status_agreed_sending = 8	Документ погоджений, відправляється
gohub_gu46_status_confirmed_paper = 9	Документ підтверджений товарним касиром
gohub_gu46_status_paper = 60	Паперовий документ



#### 4.3.4. Стани пам'ятки про подачу / прибирання вагонів і видачу / прийом контейнерів ГУ-45

Коди стану пам'яток про подачу / прибирання вагонів і видачу / прийом контейнерів ГУ-45 в *Бібліотеці* представлені перерахуванням (*enum*) `GohubGU45Status`, що містить набір констант.

*Зведена таблиця кодів станів пам'ятки про подачу / прибирання вагонів і видачу / прийом контейнерів ГУ-45*

<code>gohub_gu45_status_unknown = -1</code>	Статус невідомий
<code>gohub_gu45_status_confirmed = 2</code>	Документ узгоджений та підписаний
<code>gohub_gu45_status_canceled = 3</code>	Документ скасований
<code>gohub_gu45_status_confirmed_paper = 9</code>	Документ підтверджений товарним касиром
<code>gohub_gu45_status_paper = 60</code>	Паперовий документ

#### 4.4. Основні функції

Більш детальний опис програмного інтерфейсу бібліотеки можна отримати з файлу `gohub.client.h` (Додаток А).

##### 4.4.1. Робота з кодовими сторінками

<code>gohub_set_codepage</code>	Встановити кодування по її числовому позначенню
<code>gohub_encoding_codepage</code>	Отримати кодування по її назві
<code>gohub_encoding_codepage_w</code>	Отримати кодування по її назві (параметр в UTF-16)

##### ***gohub\_set\_codepage***

Встановити поточну кодову сторінку для всіх малих параметрів (*крім тих, що в UTF-16*).

Об'явлення:

```
int gohub_set_codepage(int codepage);
```

Параметри:

`codepage`

Кодування сторінки;

Результат:

Встановлює кодувальну сторінку для переданих рядком даних.

##### ***gohub\_encoding\_codepage***

Отримати цифрове позначення кодової сторінки по її текстовому назвою.

Об'явлення:

```
int gohub_encoding_codepage(const char* encodingName);
```

Параметри:

`encodingName`

Назва кодової сторінки;

Результат:

У разі успіху - Повертає цифрове кодування, що позначає запитану сторінку. У разі помилки - повертає нуль. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_encoding\_codepage\_w***

Отримати цифрове позначення кодової сторінки по її текстовому назвою в кодуванні UTF-16.

Об'явлення:

```
int gohub_encoding_codepage_w(  
    const GohubWChar* encodingName);
```

Параметри:

encodingName  
Назва кодової сторінки;

Результат:

У разі успіху - Повертає цифрове кодування, що позначає запитану сторінку. У разі помилки - повертає нуль. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

#### 4.4.2. Підключення до Модуля Узгодження

gohub_connect	Створити з'єднання з <i>Модулем Узгодження</i>
gohub_connect_w	Створити з'єднання з <i>Модулем Узгодження</i> (параметр в UTF-16)
gohub_disconnect	Закрити з'єднання з <i>Модулем Узгодження</i>

##### **gohub\_connect**

Встановити з'єднання з *Модулем Узгодження*.

Об'явлення:

```
GohubConnection* gohub_connect(  
    const char* host,  
    int port);
```

Параметри:

host

DNS-ім'я або IP адресаа комп'ютера, на якому встановлений *Модуль Узгодження*;

port

Номер порту для встановлення TCP з'єднання з *Модулем Узгодження*.

Результат:

У разі успіху - Показчик з'єднання з Модулем Узгодження . У разі помилки - нульовий показчик. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error та ін .

Примітки:

Після закінчення роботи зі з'єднанням, його необхідно закрити за допомогою функції gohub\_disconnect .

##### **gohub\_connect\_w**

Встановити з'єднання з *Модулем Узгодження*.

Об'явлення:

```
GohubConnection* gohub_connect(  
    const GohubWChar* host,  
    int port);
```

Параметри:

host

DNS-ім'я або IP адресаа комп'ютера, на якому встановлений *Модуль Узгодження* (в UTF-16);

port

Номер порту для встановлення TCP з'єднання з *Модулем Узгодження* .

Результат:

У разі успіху - Показчик з'єднання з Модулем Узгодження . У разі помилки - нульовий показчик. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error та ін .

Примітки:

Після закінчення роботи зі з'єднанням, його необхідно закрити за допомогою функції `gohub_disconnect`.

## ***gohub\_disconnect***

Закрити з'єднання з Модулем Узгодження.

Об'явлення:

```
GohubBool gohub_disconnect(  
    GohubConnection* connection);
```

Параметри:

```
connection  
    Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції  
    gohub_connect;
```

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### **4.4.3. Робота з документами**

<code>gohub_load_document</code>	Завантажити документ
<code>gohub_load_document_w</code>	Завантажити документ (в UTF-16)
<code>gohub_create_document</code>	Створити документ
<code>gohub_create_document_w</code>	Створити документ (в UTF-16)
<code>gohub_query_document</code>	Запросити документ
<code>gohub_query_document_w</code>	Запросити документ (в UTF-16)
<code>gohub_query_next_document</code>	Запросити наступний документ
<code>gohub_document_id</code>	Запросити ID документа
<code>gohub_document_id_w</code>	Запросити ID документа (в UTF-16)
<code>gohub_document_revision</code>	Запросити ревізію документа
<code>gohub_document_text</code>	Запросити текст документа
<code>gohub_document_text_w</code>	Запросити текст документа (в UTF-16)
<code>gohub_document_data_text</code>	Запросити текст електронних даних документа
<code>gohub_document_data_text_w</code>	Запросити текст електронних даних документа (в UTF-16)
<code>gohub_document_status</code>	Запросити статус документа
<code>gohub_document_size</code>	Запросити передбачуваний розмір документа представлений в поточній кодової сторінці
<code>gohub_document_measure equip_num</code>	Отримання відомостей вагонівимірювальної техніки
<code>gohub_document_measure equip_num_w</code>	Отримання відомостей вагонівимірювальної техніки (в UTF-16)
<code>gohub_document_set_measure equip_num_w</code>	Встановлення відомостей вагонівимірювальної техніки
<code>gohub_document_set_verified_empty_weight_for_wagon</code>	Встановлення уточненої ваги тари вагона
<code>gohub_document_get_verified_empty_weight_for_wagon</code>	Отримання уточненої ваги тари вагона
<code>gohub_document_business_unit_num</code>	Отримання номера філії ПрАТ УЗ
<code>gohub_document_business_unit_num_w</code>	Отримання номера філії ПрАТ УЗ (в UTF-16)
<code>gohub_document_set_business_unit_num</code>	Встановлення номера філії ПрАТ УЗ
<code>gohub_document_set_business_unit_num_w</code>	Встановлення номера філії ПрАТ УЗ (в UTF-16)
<code>gohub_document_get_foreign_not_accept</code>	Отримання відмітки повернення

	неприйнятих прикордонними станціями іноземних залізниць вагонів на територію України
gohub_send_document	Надіслати документ
gohub_save_document	Зберегти документ
gohub_save_document_w	Зберегти документ (в UTF-16)
gohub_save_document_data	Зберегти електронні дані документа
gohub_save_document_data_w	Зберегти електронні дані документа (в UTF-16)
gohub_close_document	Закрити документ
gohub_reclaim_document	Відкликати документ
gohub_reclaim_document_w	Відкликати документ (в UTF-16)
gohub_delete_document	Видалити документ
gohub_delete_document_w	Видалити документ (в UTF-16)
gohub_send_received_document	Надіслати документ по прибуттю
gohub_send_received_document_w	Надіслати документ по прибуттю (в UTF-16)
gohub_document_get_otpr	Зберегти актуальний текст документа
gohub_document_get_otpr_w	Зберегти актуальний текст документа (у кодуванні UTF-8)
gohub_document_get_otpr_string	Отримати актуальний текст перевізного документа
gohub_document_get_otpr_string_w	Отримати актуальний текст перевізного документа (у кодуванні UTF-8)
gohub_document_warning	Отримати текст попередження після відправки документу
gohub_document_warning_w	Отримати текст попередження після відправки документу (у кодуванні UTF-8)

## ***gohub\_load\_document***

Створення проєкта документа ( GohubDocument ) з xml-файлу.

Об'явлення:

```
GohubDocument* gohub_load_document(
    const char* path);
```

Параметри:

path

шлях до xml-файлу, з якого завантажується документ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_load\_document\_w***

Створення проєкта документа ( GohubDocument ) з xml-файлу.

Об'явлення:

```
GohubDocument* gohub_load_document_w(
    const GohubWChar* path);
```

Параметри:

path

шлях до xml-файлу в кодуванні UTF-16, з якого завантажується документ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_create\_document***

Створення про'єкта документа ( `GohubDocument` ) з рядка з xml-структурою.

Об'явлення:

```
GohubDocument* gohub_create_document(  
    const char* content);
```

Параметри:

`content`  
рядок з xml-структурою з вмістом документа;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_create\_document\_w***

Створення про'єкта документа ( `GohubDocument` ) з рядка з xml-структурою.

Об'явлення:

```
GohubDocument* gohub_create_document_w(  
    const GohubWChar* content);
```

Параметри:

`content`  
рядок з xml-структурою з вмістом документа у кодуванні UTF-16;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_document***

Запит документа з сервера АС «Клієнт УЗ» за його ID.

Об'явлення:

```
GohubDocument* gohub_query_document(  
    GohubConnection* connection,  
    const char* documentId);
```

Параметри:

`connection`  
Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`documentId`  
унікальний ідентифікатор документа, що запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_query\_document\_w***

Запит документа з сервера АС «Клієнт УЗ» за його ID.

Об'явлення:

```
GohubDocument* gohub_query_document_w(  
    GohubConnection* connection,  
    const GohubWChar* documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentId

унікальний ідентифікатор документа (в кодуванні UTF-16), який запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_query\_next\_document***

Запит документа з сервера АС «Клієнт УЗ» наступного за списком ревізій, починаючи від ревізії переданої параметром.

Об'явлення:

```
GohubDocument* gohub_query_next_document(  
    GohubConnection* connection,  
    int lastRevision);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

lastRevision

ревізія документа, від якої починається пошук наступного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_document\_id***

Запит унікального ID документа з сервера АС «Клієнт УЗ»..

Об'явлення:

```
GohubDocument* gohub_query_document(  
    GohubConnection* connection,  
    const char* documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentId

унікальний ідентифікатор документа, що запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_id\_w***

Запит документа з сервера АС «Клієнт УЗ» за його ID.

Об'явлення:

```
GohubDocument* gohub_query_document_w(  
    GohubConnection* connection,  
    const GohubWChar* documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

documentId

унікальний ідентифікатор документа (в кодуванні UTF-16), який запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_revision***

Запит ревізії документа.

Об'явлення:

```
int gohub_document_revision(  
    GohubDocument* document);
```

Параметри:

document

Об'єкт документа ( `GohubDocument`), ревізія якого запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_text***

Запит тексту документа з про'єкта документ ( `GohubDocument` ).

Об'явлення:

```
const char* gohub_document_text(  
    GohubDocument* document);
```

Параметри:

document

Об'єкт документа ( `GohubDocument` ), текст якого запитується;

Результат:



У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_document\_text\_w***

Запит тексту документа з про'єкта документ (`GohubDocument`) у кодуванні UTF-16.

Об'явлення:

```
const GohubWChar* gohub_document_text_w(  
    GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`), текст якого запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_document\_data\_text***

Запит тексту електронних даних документа з про'єкта документ (`GohubDocument`) в заданій версії ЕПД.

Об'явлення:

```
const char* gohub_document_text(  
    GohubDocument* document,  
    int epdVersion);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`), текст електронних даних якого запитується;

`epdVersion`

версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4);

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_document\_data\_text\_w***

Запит тексту електронних даних документа (у кодуванні UTF-16) з про'єкту документ (`GohubDocument`) в заданій версії ЕПД.

Об'явлення:

```
const GohubWChar* gohub_document_text_w(  
    GohubDocument* document,  
    int epdVersion);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`), текст електронних даних якого запитується;

`epdVersion`

версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4);

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_document\_status***

Запит статусу документа.

Об'явлення:

```
GohubDocumentStatus gohub_document_status(  
    GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`) , статус якого запитується;

Результат:

У разі успіху - значення з перерахування `GohubDocumentStatus` . У разі помилки - «-1». Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_document\_size***

Запит передбачуваного розміру файлу, який вийде при збереженні документа у поточній кодуванні (`GohubDocument`) .

Об'явлення:

```
int gohub_document_size(  
    GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`) , текст якого запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_document\_measure\_equip\_num***

Отримання відомостей вагонівимірювальної техніки

Об'явлення:

```
const char* gohub_document_measure_equip_num(  
    GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`) , що відправляється на сервер;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_document\_measure\_equip\_num\_w***

Отримання відомостей вагонівимірювальної техніки у кодуванні UTF-16.

Об'явлення:

```
const GohubWChar* gohub_document_measure_equip_num_w(  
    GohubDocument* document);
```

Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_document\_set\_measure\_equip\_num***

Встановлення відомостей вагонівимірювальної техніки.

Об'явлення:

```
GohubBool gohub_document_set_measure_equip_num(  
    GohubDocument* document, const char* value);
```

Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

value

значення відомостей вагонівимірювальної техніки, що потрібно встановити;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_document\_set\_measure\_equip\_num\_w***

Встановлення відомостей вагонівимірювальної техніки у кодуванні UTF-16.

Об'явлення:

```
GohubBool gohub_document_set_measure_equip_num_w(  
    GohubDocument* document, const GohubWChar* value);
```

Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

value

значення відомостей вагонівимірювальної техніки, що потрібно встановити (у кодуванні UTF-16);

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_document\_get\_verified\_empty\_weight\_for\_wagon***

Отримання уточненої ваги тари вагона

Об'явлення:

```
int gohub_document_get_verified_empty_weight_for_wagon (  
    GohubDocument* doc, int wagonIndex);
```

Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

wagonIndex

Індекс вагона уточнену вагу тари якого необхідно отримати. Індксація вагонів починається з нуля;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_document\_set\_verified\_empty\_weight\_for\_wagon***

Встановлення уточненої ваги тари вагона

Об'явлення:

```
GohubBool gohub_document_set_verified_empty_weight_for_wagon (  
    GohubDocument* doc, int wagonIndex, int weight);
```

Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

wagonIndex

Індекс вагона уточнену вагу тари якого необхідно змінити. Індксація вагонів починається з нуля;

weight

Уточнена вага тари що необхідно встановити

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_document\_warrant\_type***

Дозволяє визначити тип підстав для отримання вантажу

Об'явлення:

```
int gohub_document_warrant_type (GohubDocument* document);
```

Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - -1. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_document\_set\_warrant\_type***

Дозволяє задати тип підстав для отримання вантажу

Об'явлення:

```
GohubBool gohub_document_set_warrant_type(GohubDocument* document,
int val);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`) , що відправляється на сервер;

`val`

Тип підстав для отримання вантажу (0 - довіреність, 1 - наказ)

Результат:

У разі успіху - `true` . У разі помилки - `false` . Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_document\_business\_unit\_num***

Отримання номера філії ПрАТ УЗ

Об'явлення:

```
const char* gohub_document_business_unit_num(
GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`) , що відправляється на сервер;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_document\_business\_unit\_num\_w***

Отримання номера філії ПрАТ УЗ у кодуванні UTF-16.

Об'явлення:

```
const GohubWChar* gohub_document_business_unit_num_w(
GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`) , що відправляється на сервер;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_document\_set\_business\_unit\_num***

Встановлення номера філії ПрАТ УЗ.

**Об'явлення:**

```
GohubBool gohub_document_set_business_unit_num(  
    GohubDocument* document, const char* value);
```

**Параметри:**

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

value

значення номера філії ПрАТ УЗ, що встановлюється;

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_document\_set\_business\_unit\_num\_w***

Встановлення номера філії ПрАТ УЗ у кодуванні UTF-16.

**Об'явлення:**

```
GohubBool gohub_document_set_business_unit_num_w(  
    GohubDocument* document, const GohubWChar* value);
```

**Параметри:**

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

value

значення номера філії ПрАТ УЗ, що встановлюється (у кодуванні UTF-16);

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_document\_get\_foreign\_not\_accept***

Отримання відмітки повернення неприйнятих прикордонними станціями іноземних залізниць вагонів на територію України

**Об'явлення:**

```
bool gohub_document_get_foreign_not_accept(GohubDocument* document);
```

**Параметри:**

document

Об'єкт документа (GohubDocument)

**Результат:**

У разі успіху - значення true або false . У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_send\_document***

Відправлення документа (GohubDocument) на сервер АС «Клієнт УЗ» і далі - товарному касиру.

#### Об'явлення:

```
GohubBool gohub_send_document(  
    GohubConnection* connection,  
    GohubDocument* document);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_save\_document***

Збереження документа (GohubDocument) на диск, за вказаним шляхом та зі зазначеним кодуванням.

#### Об'явлення:

```
GohubBool gohub_save_document(  
    GohubDocument* document,  
    const char* path,  
    int codePage);
```

#### Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

path

шлях, куди зберігається файл;

codePage

числове позначення сторінки кодування, в якій зберігається файл;

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_save\_document\_w***

Збереження документа (GohubDocument) на диск, за вказаним шляхом та зі зазначеним кодуванням.

#### Об'явлення:

```
GohubBool gohub_save_document_w(  
    GohubDocument* document,  
    const GohubWChar* path,  
    int codePage);
```

#### Параметри:

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

path

шлях, куди зберігається файл (у кодуванні UTF-16);

codePage

числове позначення сторінки кодування, в якій зберігається файл;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_save\_document\_data***

Збереження електронних даних документа (`GohubDocument`) на диск, за вказаним шляхом та зі зазначеним кодуванням.

Об'явлення:

```
GohubBool gohub_save_document_data(  
    GohubDocument* document,  
    const char* path,  
    int codePage,  
    int epdVersion);
```

Параметри:

document

Об'єкт документа (`GohubDocument`), електронні дані якого запитуються;

path

шлях, куди зберігається файл;

codePage

числове позначення сторінки кодування, в якій зберігається файл;

epdVersion

версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4);

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_save\_document\_data\_w***

Збереження електронних даних документа (`GohubDocument`) на диск, за вказаним шляхом та зі зазначеним кодуванням.

Об'явлення:

```
GohubBool gohub_save_document_data_w(  
    GohubDocument* document,  
    const GohubWChar* path,  
    int codePage,  
    int epdVersion);
```

Параметри:

document



Об'єкт документа (`GohubDocument`) , електронні дані якого запитуються;

`path`

шлях, куди зберігається файл (у кодуванні UTF-16);

`codePage`

числове позначення сторінки кодування, в якій зберігається файл;

`epdVersion`

версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4);

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_close\_document***

Закриття документа (`GohubDocument`) .

Об'явлення:

```
GohubBool gohub_close_document(  
    GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_reclaim\_document***

Відкликання документа.

Об'явлення:

```
GohubBool gohub_reclaim_document(  
    GohubConnection* connection,  
    const char* documentId);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`documentId`

унікальний ідентифікатор документа, що запитується;

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_reclaim\_document\_w***

Відкликання документа.

Об'явлення:

```
GohubBool gohub_reclaim_document_w(  
    GohubConnection* connection,  
    const GohubWChar* documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentId

унікальний ідентифікатор документа (в кодуванні UTF-16), який запитується;

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_delete\_document***

Видалення документа.

Об'явлення:

```
GohubBool gohub_delete_document(  
    GohubConnection* connection,  
    const char* documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentId

унікальний ідентифікатор документа, що запитується.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_delete\_document\_w***

Видалення документа.

Об'явлення:

```
GohubBool gohub_delete_document_w(  
    GohubConnection* connection,  
    const GohubWChar* char documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentId

унікальний ідентифікатор документа (в кодуванні UTF-16), який запитується.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_send\_received\_document***

Відправлення документа по прибуттю (GohubDocument) на сервер АС «Клієнт УЗ» і далі - товарному касиру.

Об'явлення:

```
GohubBool gohub_send_document(  
    GohubConnection* connection,  
    GohubDocument* document  
    const char* documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

documentId

унікальний ідентифікатор документа (у кодуванні UTF-16), що відправляється.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_send\_received\_document\_w***

Відправлення документа по прибуттю (GohubDocument) на сервер АС «Клієнт УЗ» і далі - товарному касиру.

Об'явлення:

```
GohubBool gohub_send_document(  
    GohubConnection* connection,  
    GohubDocument* document  
    const GohubWChar* char documentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

document

Об'єкт документа (GohubDocument) , що відправляється на сервер;

documentId

унікальний ідентифікатор документа (у кодуванні UTF-16), що відправляється.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_get\_otpr***

Збереження актуального тексту перевізного документа (`GohubDocument`) у форматі `xml` на диск, за вказаним шляхом.

Об'явлення:

```
GohubBool gohub_document_get_otpr(  
    GohubDocument* document,  
    const char* path);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`), електронні дані якого запитуються;

`path`

шлях, куди зберігається файл;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_get\_otpr\_w***

Збереження актуального тексту перевізного документа (`GohubDocument`) у форматі `xml` на диск, за вказаним шляхом (у кодуванні UTF-8).

Об'явлення:

```
GohubBool gohub_document_get_otpr_w(  
    GohubDocument* document,  
    const GohubWChar* path);
```

Параметри:

`document`

Об'єкт документа (`GohubDocument`), електронні дані якого запитуються;

`path`

шлях, куди зберігається файл (у кодуванні UTF-16);

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_get\_otpr\_string***

Дозволяє отримати актуальний текст перевізного документа (`GohubDocument`) у вигляді рядку.

Об'явлення:

```
const char* gohub_document_get_otpr_string(GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (GohubDocument) , електронні дані якого запитуються;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_get\_otpr\_string\_w***

Дозволяє отримати актуальний текст перевізного документа (GohubDocument) у вигляді рядку (у кодуванні UTF-8).

Об'явлення:

```
GohubBool const GohubWChar* gohub_document_get_otpr_string_w(  
GohubDocument* document);
```

Параметри:

document

Об'єкт документа (GohubDocument) , електронні дані якого запитуються;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_warning***

Дозволяє отримати актуальний текст попереджень перевізного документа (GohubDocument) у вигляді рядку після відправки документа.

Об'явлення:

```
const char* gohub_document_warning(GohubDocument* document);
```

Параметри:

document

Об'єкт документа (GohubDocument) , електронні дані якого запитуються;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_warning\_w***

Дозволяє отримати актуальний текст попереджень перевізного документа (GohubDocument) у вигляді рядку після відправки документа (у кодуванні UTF-8).

Об'явлення:

```
const GohubWChar* gohub_document_warning_w(GohubDocument* document);
```

Параметри:

document

Об'єкт документа (GohubDocument) , електронні дані якого запитуються;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

#### 4.4.4. Робота зі супровідними документами

gohub_load_attachment	Завантажити супровідний документ, для бланка ГУ та ЦІМ
gohub_load_attachment_with_user_data	Завантаження супровідного документа (для бланків ГУ та ЦІМ) з електронними даними користувача.
gohub_load_attachment_w	Завантажити супровідний документ для бланка ГУ та ЦІМ (в UTF-16)
gohub_load_attachment_with_user_data_w	Завантаження супровідного документа (для бланків ГУ та ЦІМ) з електронними даними користувача (в UTF-16)
gohub_load_smgs_attachment	Завантажити супровідний документ для бланка СМГС та ЦІМ/СМГС
gohub_load_smgs_attachment_with_user_data	Завантаження супровідного документа (для бланків СМГС та ЦІМ/СМГС) з електронними даними користувача.
gohub_load_smgs_attachment_w	Завантажити супровідний документ для бланка СМГС та ЦІМ/СМГС (в UTF-16)
gohub_load_smgs_attachment_with_user_data_w	Завантаження супровідного документа (для бланків СМГС та ЦІМ/СМГС) з електронними даними користувача. (в UTF-16)
gohub_send_attachment	Відправити супровідний документ
gohub_query_attachment	Запитати супровідний документ
gohub_query_attachment_w	Запитати супровідний документ (в UTF-16)
gohub_save_attachment	Зберегти супровідний документ
gohub_save_attachment_with_user_data	Збереження електронних даних супровідного документа користувача
gohub_save_attachment_w	Зберегти супровідний документ (в UTF-16)
gohub_save_attachment_with_user_data_w	Збереження електронних даних супровідного документа користувача (в UTF-16).
gohub_delete_attachment	Видалити супровідний документ
gohub_delete_attachment_w	Видалити супровідний документ (в UTF-16)
gohub_close_attachment	Закрити супровідний документ
gohub_attachment_id	Запитати ID супровідного документа
gohub_attachment_id_w	Запитати ID супровідного документа (в UTF-16)
gohub_attachment_type_code	Запитати код типу супровідного документа
gohub_attachment_type_code_w	Запитати код типу супровідного документа (в UTF-16)
gohub_attachment_name	Запитати ім'я супровідного документа
gohub_attachment_name_w	Запитати ім'я супровідного документа (в UTF-16)
gohub_attachment_reg_number	Запитати реєстраційний номер супровідного документа
gohub_attachment_reg_number_w	Запитати реєстраційний номер супровідного документа (в UTF-16)
gohub_attachment_reg_date	Запитати дату реєстрації супровідного документа
gohub_attachment_reg_date_w	Запитати дату реєстрації супровідного документа (в UTF-16)
gohub_attachment_valid_from	Запитати дату початку дії супровідного документа
gohub_attachment_valid_from_w	Запитати дату початку дії супровідного документа (в UTF-16)
gohub_attachment_valid_to	Запитати дату припинення дії супровідного документа
gohub_attachment_valid_to_w	Запитати дату припинення дії супровідного документа (в UTF-16)
gohub_attachment_description	Запитати дескриптор супровідного документа
gohub_attachment_description_w	Запитати дескриптор супровідного документа в UTF-16)
gohub_attachment_count	Запитати кількість доданих до перевізного

	документу супровідних документів
gohub_attachment_id_by_index	Запитати ID супровідного документа, за його індексом у переліку доданих до перевізного документа супровідних документів
gohub_attachment_id_by_index_w	Запитати ID супровідного документа, за його індексом у переліку доданих до перевізного документа супровідних документів (в UTF-16)

## ***gohub\_load\_attachment***

Завантаження супровідного документа (для бланків ГУ та ЦІМ).

Об'явлення:

```
GohubAttachment* gohub_load_attachment(
    const char* typeCode,
    const char* name,
    const char* regNumber,
    const char* regDate,
    const char* validFrom,
    const char* validTo,
    const char* path);
```

Параметри:

typeCode

Код типу супровідного документа;

name

ім'я супровідного документа;

regNumber

реєстраційний номер супровідного документа;

regDate

дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy');

validFrom

дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

validTo

дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

path

шлях до файлу з відсканованим текстом супровідного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_load\_attachment\_with\_user\_data***

Завантаження супровідного документа (для бланків ГУ та ЦІМ) з електронними даними користувача.

#### Об'явлення:

```
GohubAttachment* gohub_load_attachment_with_user_data(  
    const char* typeCode,  
    const char* name,  
    const char* regNumber,  
    const char* regDate,  
    const char* validFrom,  
    const char* validTo,  
    const char* path ,  
    const char* pathUserData);
```

#### Параметри:

typeCode

Код типу супровідного документа;

name

ім'я супровідного документа;

regNumber

реєстраційний номер супровідного документа;

regDate

дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy');

validFrom

дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

validTo

дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

path

шлях до файлу з відсканованим текстом супровідного документа.

pathUserData

шлях до файлу з електронними даними супровідного документа.

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і `in`.

### ***gohub\_load\_attachment\_w***

Завантаження супровідного документа (для бланків ГУ та ЦІМ) (в UTF-16).

#### Об'явлення:

```
GohubAttachment* gohub_load_attachment_w(  
    const GohubWChar* typeCode,  
    const GohubWChar* name,  
    const GohubWChar* regNumber,  
    const GohubWChar* regDate,  
    const GohubWChar* validFrom,  
    const GohubWChar* validTo,  
    const GohubWChar* path);
```



#### Параметри:

typeCode

Код типу супровідного документа у кодуванні UTF-16;

name

Ім'я супровідного документа у кодуванні UTF-16;

regNumber

Реєстраційний номер супровідного документа у кодуванні UTF-16;

regDate

Дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validFrom

Дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validTo

Дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

path

Шлях до файлу з відсканованим текстом супровідного документа у кодуванні UTF-16.

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_load\_attachment\_with\_user\_data\_w***

Завантаження супровідного документа (для бланків ГУ та ЦІМ) з електронними даними користувача (в UTF-16).

#### Об'явлення:

```
GohubAttachment* gohub_load_attachment_with_user_data_w(  
    const GohubWChar* typeCode,  
    const GohubWChar* name,  
    const GohubWChar* regNumber,  
    const GohubWChar* regDate,  
    const GohubWChar* validFrom,  
    const GohubWChar* validTo,  
    const GohubWChar* path,  
    const GohubWChar* pathUserData);
```

#### Параметри:

typeCode

Код типу супровідного документа у кодуванні UTF-16;

name

ім'я супровідного документа у кодуванні UTF-16;

regNumber

реєстраційний номер супровідного документа у кодуванні UTF-16;

regDate

дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validFrom

дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validTo

дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

path

шлях до файлу з відсканованим текстом супровідного документа у кодуванні UTF-16.

pathUserData

шлях до файлу з електронними даними супровідного документа у кодуванні UTF-16.

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_load\_smgs\_attachment***

Завантаження супровідного документа (для бланків СМГС та ЦІМ/СМГС).

#### Об'явлення:

```
GohubAttachment* gohub_load_smgs_attachment(  
    const char* smgsTypeCode,  
    const char* name,  
    const char* regNumber,  
    const char* regDate,  
    const char* validFrom,  
    const char* validTo,  
    const char* path);
```

#### Параметри:

smgsTypeCode

Код типу супровідного документа згідно інформаційного керівництва СМГС;

name

Ім'я супровідного документа;

regNumber

Реєстраційний номер супровідного документа;

regDate

Дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy');

validFrom

Дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

validTo

Дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

path

Шлях до файлу з відсканованим текстом супровідного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_load\_smgs\_attachment\_with\_user\_data***

Завантаження супровідного документа (для бланків СМГС та ЦІМ/СМГС) з електронними даними користувача.

Об'явлення:

```
GohubAttachment* gohub_load_smgs_attachment_with_user_data(  
    const char* smgsTypeCode,  
    const char* name,  
    const char* regNumber,  
    const char* regDate,  
    const char* validFrom,  
    const char* validTo,  
    const char* path,  
    const char* pathUserData);
```

Параметри:

smgsTypeCode

Код типу супровідного документа згідно інформаційного керівництва СМГС;

name

Ім'я супровідного документа;

regNumber

Реєстраційний номер супровідного документа;

regDate

Дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy');

validFrom

Дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

validTo

Дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

path

Шлях до файлу з відсканованим текстом супровідного документа.

pathUserData

Шлях до файлу з електронними даними супровідного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_load\_smgs\_attachment\_w***

Завантаження супровідного документа (для бланків СМГС та ЦІМ/СМГС) в UTF-16.

Об'явлення:

```
GohubAttachment* gohub_load_smgs_attachment_w(  
    const GohubWChar* smgsTypeCode,  
    const GohubWChar* name,  
    const GohubWChar* regNumber,  
    const GohubWChar* regDate,  
    const GohubWChar* validFrom,  
    const GohubWChar* validTo,  
    const GohubWChar* path);
```

Параметри:

`smgsTypeCode`

Код типу супровідного документа у кодуванні UTF-16 згідно інформаційного керівництва СМГС;

`name`

Ім'я супровідного документа у кодуванні UTF-16;

`regNumber`

Реєстраційний номер супровідного документа у кодуванні UTF-16;

`regDate`

Дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

`validFrom`

Дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

`validTo`

Дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

`path`

Шлях до файлу з відсканованим текстом супровідного документа у кодуванні UTF-16.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_load\_smgs\_attachment\_with\_user\_data\_w***

Завантаження супровідного документа (для бланків СМГС та ЦІМ/СМГС) з електронними даними користувача (в UTF-16).

#### Об'явлення:

```
GohubAttachment* gohub_load_smgs_attachment_with_user_data_w(  
    const GohubWChar* smgsTypeCode,  
    const GohubWChar* name,  
    const GohubWChar* regNumber,  
    const GohubWChar* regDate,  
    const GohubWChar* validFrom,  
    const GohubWChar* validTo,  
    const GohubWChar* path,  
    const GohubWChar* pathUserData);
```

#### Параметри:

smgsTypeCode

Код типу супровідного документа у кодуванні UTF-16 згідно інформаційного керівництва СМГС;

name

Ім'я супровідного документа у кодуванні UTF-16;

regNumber

Реєстраційний номер супровідного документа у кодуванні UTF-16;

regDate

Дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validFrom

Дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validTo

Дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

path

Шлях до файлу з відсканованим текстом супровідного документа у кодуванні UTF-16.

pathUserData

Шлях до файлу з електронними даними супровідного документа у кодуванні UTF-16.

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_send\_attachment***

Відправлення супровідного документа.

#### Об'явлення:

```
GohubBool gohub_send_attachment(  
    GohubConnection* connection,
```

```
GohubAttachment* attachment);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

attachment

Об'єкт документа (GohubAttachment) , що відправляється на сервер.

**Результат:**

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_attachment***

Запит супровідного документа.

**Об'явлення:**

```
GohubAttachment* gohub_query_attachment(  
    GohubConnection* connection,  
    const char* attachmentId);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

attachmentId

унікальний ідентифікатор супровідного документа.

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_attachment\_w***

Запит супровідного документа.

**Об'явлення:**

```
GohubAttachment* gohub_query_attachment_w(  
    GohubConnection* connection,  
    const GohubWChar* attachmentId);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

attachmentId

унікальний ідентифікатор супровідного документа (у кодуванні UTF-16).

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_attachment\_with\_user\_data***

Запит супровідного документа з електронними даними користувача.

Об'явлення:

```
GohubAttachment* gohub_query_attachment_with_user_data(  
    GohubConnection* connection,  
    const char* attachmentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

attachmentId

унікальний ідентифікатор супровідного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_query\_attachment\_with\_user\_data\_w***

Запит супровідного документа.

Об'явлення:

```
GohubAttachment* gohub_query_attachment_with_user_data_w(  
    GohubConnection* connection,  
    const GohubWChar* attachmentId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

attachmentId

унікальний ідентифікатор супровідного документа (у кодуванні UTF-16).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_save\_attachment***

Збереження супровідного документа.

Об'явлення:

```
GohubBool gohub_save_attachment(  
    GohubAttachment* attachment,  
    const char* path);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment), що зберігається на диск;

path

шлях, куди зберігається файл.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_save\_attachment\_with\_user\_data***

Збереження електронних даних супровідного документа користувача.

Об'явлення:

```
GohubBool gohub_save_attachment_with_user_data(  
    GohubAttachment* attachment,  
    const char* path);
```

Параметри:

`attachment`

Об'єкт документа (`GohubAttachment`), електронні дані якого зберігаються на диск;

`path`

Шлях, куди зберігається файл.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_save\_attachment\_w***

Збереження супровідного документа (в UTF-16).

Об'явлення:

```
GohubBool gohub_save_attachment_w(  
    GohubAttachment* attachment,  
    const GohubWChar* path);
```

Параметри:

`attachment`

Об'єкт документа (`GohubAttachment`), що зберігається на диск;

`path`

Шлях, куди зберігається файл (у кодуванні UTF-16).

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_save\_attachment\_with\_user\_data\_w***

Збереження електронних даних супровідного документа користувача (в UTF-16).

Об'явлення:

```
GohubBool gohub_save_attachment_with_user_data_w(  
    GohubAttachment* attachment,  
    const GohubWChar* path);
```

Параметри:

`attachment`



Об'єкт документа (`GohubAttachment`), електронні дані якого зберігаються на диск;

`path`

Шлях, куди зберігається файл (у кодуванні UTF-16).

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_delete\_attachment***

Видалення супровідного документа.

Об'явлення:

```
GohubBool gohub_delete_attachment(  
    GohubConnection* connection,  
    const char* attachmentId);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`attachmentId`

Унікальний ідентифікатор супровідного документа, що видаляється.

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_delete\_attachment\_w***

Запит супровідного документа.

Об'явлення:

```
GohubBool gohub_delete_attachment_w(  
    GohubConnection* connection,  
    const GohubWChar* attachmentId);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`attachmentId`

унікальний ідентифікатор супровідного документа (у кодуванні UTF-16), що видаляється.

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_close\_attachment***

Закриття супровідного документа.

Об'явлення:

```
GohubBool gohub_close_attachment(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_attachment\_id***

Запит ID супровідного документа.

Об'явлення:

```
const char* gohub_attachment_id(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_attachment\_id\_w***

Запит ID супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_id_w(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_attachment\_type\_code***

Запит кода типу супровідного документа.

Об'явлення:

```
const char* gohub_attachment_type_code(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_attachment\_type\_code\_w***

Запит кода типу супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_type_code_w(  
    GohubAttachment* attachment);
```

Параметри:

`attachment`

Об'єкт документа (`GohubAttachment`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_attachment\_name***

Запит імені супровідного документа.

Об'явлення:

```
const char* gohub_attachment_name(  
    GohubAttachment* attachment);
```

Параметри:

`attachment`

Об'єкт документа (`GohubAttachment`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_attachment\_name\_w***

Запит імені супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_name_w(  
    GohubAttachment* attachment);
```

Параметри:

`attachment`

Об'єкт документа (`GohubAttachment`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_attachment\_reg\_number***

Запит реєстраційного номера супровідного документа.

Об'явлення:

```
const char* gohub_attachment_reg_number(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_attachment\_reg\_number\_w***

Запит реєстраційного номера супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_reg_number_w(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_attachment\_reg\_date***

Запит дати реєстрації супровідного документа.

Об'явлення:

```
const char* gohub_attachment_reg_date(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_attachment\_reg\_date\_w***

Запит дати реєстрації супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_reg_date_w(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_attachment\_valid\_from***

Запит дати початку дії супровідного документа.

Об'явлення:

```
const char* gohub_attachment_valid_from(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_attachment\_valid\_from\_w***

Запит дати початку дії супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_valid_from_w(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_attachment\_valid\_to***

Запит дати припинення дії супровідного документа.

Об'явлення:

```
const char* gohub_attachment_valid_to(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_attachment\_valid\_to\_w***

Запит дати припинення супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_valid_to_w(  
    GohubAttachment* attachment);
```

Параметри:

attachment

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_attachment\_description***

Запит дескриптора супровідного документа. Повертається рядок що містить назву типу документа, його реєстраційного номера та дати.

Об'явлення:

```
const char* gohub_attachment_description(  
    GohubAttachment* attachment);
```

Параметри:

`attachment`

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_attachment\_description\_w***

Запит дескриптора супровідного документа. Повертається рядок що містить назву типу документа, його реєстраційного номера та дати (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_description_w(  
    GohubAttachment* attachment);
```

Параметри:

`attachment`

Об'єкт документа (GohubAttachment).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_attachment\_count***

Запитати кількість доданих супровідних документів до перевізного документу.

Об'явлення:

```
int gohub_attachment_count(  
    GohubDocument* document);
```

Параметри:

`document`

Об'єкт документа (GohubDocument).

Результат:

У разі успіху - не негативна число. У разі помилки - «-1». Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_attachment\_id\_by\_index***

Запитати ID супровідного документа за його індексом у переліку доданих супровідних документів до перевізного документу.

Об'явлення:

```
const char* gohub_attachment_id_by_index(  
    GohubDocument* document,  
    int attachmentIndex);
```

Параметри:

document

Об'єкт документа (GohubDocument);

attachmentIndex

Індекс документа у переліку доданих до перевізного документу.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_attachment\_id\_by\_index\_w***

Запитати ID супровідного документа за його індексом у переліку доданих супровідних документів до перевізного документу (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_attachment_id_by_index_w(  
    GohubDocument* document,  
    int attachmentIndex);
```

Параметри:

document

Об'єкт документа (GohubDocument);

attachmentIndex

Індекс документа у переліку доданих до перевізного документу.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### **4.4.5. Робота з електронними даними і пакетами попереднього інформування (ПІ)**

gohub_load_edata	Завантажити електронні дані ПІ
gohub_load_edata_w	Завантажити електронні дані ПІ (в UTF-16)
gohub_load_edata_simple	Завантажити електронні дані ПІ без супровідного документа
gohub_load_edata_simple_w	Завантажити електронні дані ПІ без супровідного документа (в UTF-16)
gohub_send_edata	Відправити електронні дані ПІ
gohub_update_edata	Оновити наявні електронні дані ПІ
gohub_query_edata	Запитати електронні дані ПІ
gohub_query_edata_w	Запитати електронні дані ПІ (в UTF-16)
gohub_query_next_edata	Запитати електронні дані ПІ за ревізією
gohub_save_edata	Зберегти електронні дані ПІ

gohub_save_edata_w	Зберегти електронні дані ПІ (в UTF-16)
gohub_edata_load_data	Завантажити дані з файлу у наявні електронні дані ПІ
gohub_edata_load_data_w	Завантажити дані з файлу у наявні електронні дані ПІ (в UTF-16)
gohub_close_edata	Закрити електронні дані ПІ
gohub_edata_id	Запитати ID електронних даних ПІ
gohub_edata_id_w	Запитати ID електронних даних ПІ (в UTF-16)
gohub_edata_revision	Запитати ревізію електронних даних ПІ
gohub_edata_revision_date	Запитати дату ревізії електронних даних ПІ
gohub_edata_revision_date_w	Запитати дату ревізії електронних даних ПІ (в UTF-16)
gohub_edata_doc_type	Запитати код типу електронних даних ПІ (190 рахунок-фактура, 320 пакувальний лист)
gohub_edata_status	Запитати статус електронних даних ПІ
gohub_edata_version	Запитати версію електронних даних ПІ
gohub_edata_version_w	Запитати версію електронних даних ПІ (в UTF-16)
gohub_edata_attachment_id	Запитати ID супровідного документа електронних даних ПІ
gohub_edata_attachment_id_w	Запитати ID супровідного документа електронних даних ПІ (в UTF-16)
gohub_query_pi_package	Запитати пакет ПІ
gohub_query_pi_package_w	Запитати пакет ПІ (в UTF-16)
gohub_query_next_pi_package	Запитати пакет ПІ за ревізією
gohub_save_pi_package	Зберегти пакет ПІ
gohub_save_pi_package_w	Зберегти пакет ПІ (в UTF-16)
gohub_close_pi_package	Закрити пакет ПІ
gohub_pi_package_id	Запитати ID пакета ПІ
gohub_pi_package_id_w	Запитати ID пакета ПІ (в UTF-16)
gohub_pi_package_revision	Запитати ревізію пакета ПІ
gohub_pi_package_revision_date	Запитати дату ревізії пакета ПІ
gohub_pi_package_revision_date_w	Запитати дату ревізії пакета ПІ (в UTF-16)
gohub_pi_package_consignment_id	Запитати ID перевізного документа на який посилається пакет ПІ
gohub_pi_package_consignment_id_w	Запитати ID перевізного документа на який посилається пакет ПІ (в UTF-16)
gohub_pi_package_status	Запитати статус пакета ПІ
gohub_pi_package_pipacktoed_count	Запитати кількість елементів GohubPiPackageToEData що містяться у пакеті ПІ
gohub_pi_package_pipacktoed	Запитати елемент GohubPiPackageToEData у пакеті ПІ за індексом
gohub_pipacktoed_id	Запитати ID про'єкту GohubPiPackageToEData
gohub_pipacktoed_id_w	Запитати ID про'єкту GohubPiPackageToEData (в UTF-16)
gohub_pipacktoed_edata_id	Запитати ID електронних даних в про'єкті GohubPiPackageToEData
gohub_pipacktoed_edata_id_w	Запитати ID електронних даних в про'єкті GohubPiPackageToEData (в UTF-16)
gohub_pipacktoed_pi_package_id	Запитати ID пакета ПІ в про'єкті GohubPiPackageToEData
gohub_pipacktoed_pi_package_id_w	Запитати ID пакета ПІ в про'єкті GohubPiPackageToEData (в UTF-16)
gohub_pipacktoed_note	Запитати примітки в про'єкті GohubPiPackageToEData
gohub_pipacktoed_note_w	Запитати примітки в про'єкті GohubPiPackageToEData (в UTF-16)
gohub_pipacktoed_status	Запитати статус про'єкту GohubPiPackageToEData
gohub_pipacktoed_edata_version	Запитати версію електронних даних в про'єкті GohubPiPackageToEData
gohub_pipacktoed_edata_version_w	Запитати версію електронних даних в про'єкті GohubPiPackageToEData (в UTF-16)
gohub_add_edata_to_pi_package	Додати електронні дані до пакету ПІ
gohub_add_edata_to_pi_package_w	Додати електронні дані до пакету ПІ (в UTF-16)



## ***gohub\_load\_edata***

Завантаження електронних даних ПІ.

Об'явлення:

```
GohubEData* gohub_load_edata(  
    unsigned int attachmentSmgsTypeCode,  
    const char* xmlPath,  
    const char* name,  
    const char* regNumber,  
    const char* regDate,  
    const char* validFrom,  
    const char* validTo,  
    const char* pdfPath);
```

Параметри:

attachmentSmgsTypeCode

Код типу супровідного документа згідно інформаційного керівництва СМГС. Можна вибрати зі значень 325 Рахунок-проформа (інвойс), 380 (інвойс) рахунок-фактура, 935 Рахунок-фактура;

xmlPath

Шлях до xml-файлу що містять електронні дані ПІ;

name

Ім'я супровідного документа;

regNumber

Реєстраційний номер супровідного документа;

regDate

Дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy');

validFrom

Дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

validTo

Дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy');

pdfPath

Шлях до pdf-файлу з відсканованим текстом супровідного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_load\_edata\_w***

Завантаження електронних даних ПІ (у кодуванні UTF-16).

Об'явлення:

```
GohubEData* gohub_load_edata_w(  
    unsigned int attachmentSmgsTypeCode,
```

```
const GohubWChar* xmlPath,
const GohubWChar* name,
const GohubWChar* regNumber,
const GohubWChar* regDate,
const GohubWChar* validFrom,
const GohubWChar* validTo,
const GohubWChar* pdfPath);
```

#### Параметри:

attachmentSmgsTypeCode

Код типу супровідного документа згідно інформаційного керівництва СМГС. Можна вибрати зі значень 325 Рахунок-проформа (інвойс), 380 (інвойс) рахунок-фактура, 935 Рахунок-фактура;

xmlPath

Шлях до xml-файлу у кодуванні UTF-16 що містять електронні дані ПІ;

name

Ім'я супровідного документа у кодуванні UTF-16;

regNumber

Реєстраційний номер супровідного документа у кодуванні UTF-16;

regDate

Дата реєстрації супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validFrom

Дата початку терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

validTo

Дата припинення терміну дії супровідного документа (формат дати 'dd.MM.yyyy') у кодуванні UTF-16;

pdfPath

Шлях до pdf-файлу з відсканованим текстом супровідного документа у кодуванні UTF-16.

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_load\_edata\_simple***

Завантажити електронні дані ПІ без супровідного документа.

#### Об'явлення:

```
GohubEDData* gohub_load_edata(
    unsigned int attachmentSmgsTypeCode,
    const char* xmlPath
);
```

#### Параметри:

attachmentSmgsTypeCode

Код типу супровідного документа згідно інформаційного керівництва СМГС. Можна вибрати зі значень 325 Рахунок-проформа (інвойс), 380 (інвойс) рахунок-фактура, 935 Рахунок-фактура;

xmlPath

Шлях до xml-файлу що містять електронні дані ПІ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_load\_edata\_simple\_w***

Завантажити електронні дані ПІ без супровідного документа (у кодуванні UTF-16).

Об'явлення:

```
GohubEData* gohub_load_edata_w(  
    unsigned int attachmentSmgsTypeCode,  
    const GohubWChar* xmlPath);
```

Параметри:

attachmentSmgsTypeCode

Код типу супровідного документа згідно інформаційного керівництва СМГС. Можна вибрати зі значень 325 Рахунок-проформа (інвойс), 380 (інвойс) рахунок-фактура, 935 Рахунок-фактура;

xmlPath

Шлях до xml-файлу у кодуванні UTF-16 що містять електронні дані ПІ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_send\_edata***

Відправлення електронних даних ПІ.

Об'явлення:

```
GohubBool gohub_send_edata(  
    GohubConnection* connection,  
    GohubEData* eData);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

eData

Об'єкт електронних даних ПІ (`GohubEData`) , що відправляється на сервер.

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_update\_edata***

Оновити наявні електронні дані ПІ.

Об'явлення:

```
GohubBool gohub_update_edata(  
    GohubConnection* connection,  
    GohubEData* eData);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

eData

Об'єкт електронних даних ПІ (GohubEData) , що буде оновлюватись на сервері.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_edata***

Запит електронних даних ПІ.

Об'явлення:

```
GohubEData* gohub_query_edata(  
    GohubConnection* connection,  
    const char* eDataId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

eDataId

Унікальний ідентифікатор електронних даних ПІ.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_edata\_w***

Запит електронних даних ПІ (у кодуванні UTF-16).

Об'явлення:

```
GohubEData* gohub_query_edata_w(  
    GohubConnection* connection,  
    const GohubWChar* eDataId);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

eDataId

Унікальний ідентифікатор електронних даних ПІ (у кодуванні UTF-16).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_edata\_for\_attachment***

Запит електронних даних ПІ за ідентифікатором супровідного документа.

Об'явлення:

```
GohubEData* gohub_query_edata(  
    GohubConnection* connection,  
    const char* attachmentId);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`attachmentId`

Унікальний ідентифікатор супровідного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_edata\_for\_attachment\_w***

Запит електронних даних ПІ (у кодуванні UTF-16) за ідентифікатором супровідного документа.

Об'явлення:

```
GohubEData* gohub_query_edata_w(  
    GohubConnection* connection,  
    const GohubWChar* attachmentId);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`attachmentId`

Унікальний ідентифікатор супровідного документа (у кодуванні UTF-16).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_next\_edata***

Запит електронних даних ПІ за ревізією.

Об'явлення:

```
GohubEData* gohub_query_next_edata(  
    GohubConnection* connection,
```

```
unsigned __int64 lastRevision);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

lastRevision

Ревізія, з якої починати пошук електронних даних ПІ.

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_save\_edata***

Збереження електронних даних ПІ.

**Об'явлення:**

```
GohubBool gohub_save_edata(  
    GohubEData* eData,  
    const char* path);
```

**Параметри:**

eData

Об'єкт електронних даних ПІ (GohubEData) , що зберігається на диск;

path

Шлях, куди зберігається файл.

**Результат:**

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_save\_edata\_w***

Збереження електронних даних ПІ (у кодуванні UTF-16).

**Об'явлення:**

```
GohubBool gohub_save_edata_w(  
    GohubEData* eData,  
    const GohubWChar* path);
```

**Параметри:**

eData

Об'єкт електронних даних ПІ (GohubEData) , що зберігається на диск;

path

Шлях, куди зберігається файл (у кодуванні UTF-16).

**Результат:**

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_edata\_load\_data***

Завантажити дані з файлу у наявні електронні дані ПП.

Об'явлення:

```
GohubBool gohub_save_edata(  
    GohubEData* eData,  
    const char* path);
```

Параметри:

eData

Об'єкт електронних даних ПП (GohubEData) , що зберігається на диск;

path

Шлях, куди зберігається файл.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_edata\_load\_data\_w***

Завантажити дані з файлу у наявні електронні дані ПП (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_save_edata_w(  
    GohubEData* eData,  
    const GohubWChar* path);
```

Параметри:

eData

Об'єкт електронних даних ПП (GohubEData) , що зберігається на диск;

path

Шлях, куди зберігається файл (у кодуванні UTF-16).

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_close\_edata***

Закриття електронних даних ПП.

Об'явлення:

```
GohubBool gohub_close_edata(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПП (GohubEData) .

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_edata\_id***

Запит ID електронних даних ПІ.

Об'явлення:

```
const char* gohub_edata_id(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_edata\_id\_w***

Запит ID електронних даних ПІ (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_edata_id_w(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_edata\_revision***

Запит номера ревізії електронних даних ПІ.

Об'явлення:

```
unsigned __int64 gohub_edata_revision(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_edata\_revision\_date***

Запит дати ревізії електронних даних ПІ.

Об'явлення:

```
const char* gohub_edata_revision_date(  
    GohubEData* eData);
```

Параметри:

eData



Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_edata\_revision\_date\_w***

Запит дати ревізії електронних даних ПІ (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_edata_revision_date_w(  
    GohubEData* eData);
```

Параметри:

`eData`

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_edata\_doc\_type***

Запит типу електронних даних ПІ.

Об'явлення:

```
unsigned int gohub_edata_doc_type(  
    GohubEData* eData);
```

Параметри:

`eData`

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_edata\_status***

Запит статусу електронних даних ПІ.

Об'явлення:

```
int gohub_edata_status(  
    GohubEData* eData);
```

Параметри:

`eData`

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_edata\_version***

Запит версії електронних даних ПІ.

Об'явлення:

```
const char* gohub_edata_version(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_edata\_version\_w***

Запит версії електронних даних ПІ (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_edata_version_w(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_edata\_attachment\_id***

Запит ID супровідного документа електронних даних ПІ.

Об'явлення:

```
const char* gohub_edata_attachment_id(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_edata\_attachment\_id\_w***

Запит ID супровідного документа електронних даних ПІ (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_edata_attachment_id_w(  
    GohubEData* eData);
```

Параметри:

eData

Об'єкт електронних даних ПІ (GohubEData).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_query\_pi\_package***

Запит пакет ПІ.

Об'явлення:

```
GohubPiPackage* gohub_query_pi_package(  
    GohubConnection* connection,  
    const char* piPackageId);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`piPackageId`

Унікальний ідентифікатор електронних даних ПІ.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_query\_pi\_package\_w***

Запит пакет ПІ (у кодуванні UTF-16).

Об'явлення:

```
GohubPiPackage* gohub_query_pi_package_w(  
    GohubConnection* connection,  
    const GohubWChar* piPackageId);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`piPackageId`

Унікальний ідентифікатор супровідного документа (у кодуванні UTF-16).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_query\_next\_pi\_package***

Запит пакета ПІ за ревізією.

Об'явлення:

```
GohubPiPackage* gohub_query_next_pi_package(  
    GohubConnection* connection,  
    unsigned __int64 lastRevision);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`lastRevision`

Ревізія, з якої починати пошук пакета ПІ.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_save\_pi\_package***

Збереження пакета ПІ.

Об'явлення:

```
GohubBool gohub_save_pi_package(  
    GohubPiPackage* piPackage,  
    const char* path);
```

Параметри:

`piPackage`

Об'єкт пакета ПІ (`GohubPiPackage`) , що зберігається на диск;

`path`

Шлях, куди зберігається файл.

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_save\_pi\_package\_w***

Збереження пакета ПІ (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_save_pi_package_w(  
    GohubPiPackage* piPackage,  
    const GohubWChar* path);
```

Параметри:

`piPackage`

Об'єкт пакета ПІ (`GohubPiPackage`) , що зберігається на диск;

`path`

Шлях, куди зберігається файл (у кодуванні UTF-16).

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_close\_pi\_package***

Закриття пакета ПІ.

Об'явлення:

```
GohubBool gohub_close_pi_package(  

```

```
GohubPiPackage* piPackage);
```

Параметри:

```
piPackage
```

Об'єкт пакета ПІ (GohubPiPackage).

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_pi\_package\_id***

Запит ID пакета ПІ.

Об'явлення:

```
const char* gohub_pi_package_id(  
    GohubPiPackage* piPackage);
```

Параметри:

```
piPackage
```

Об'єкт пакета ПІ (GohubPiPackage).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_pi\_package\_id\_w***

Запит ID пакета ПІ (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pi_package_id_w(  
    GohubPiPackage* piPackage);
```

Параметри:

```
piPackage
```

Об'єкт пакета ПІ (GohubPiPackage).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_pi\_package\_revision***

Запит номера ревізії пакета ПІ.

Об'явлення:

```
unsigned __int64 gohub_pi_package_revision(  
    GohubPiPackage* piPackage);
```

Параметри:

```
piPackage
```

Об'єкт пакета ПІ (GohubPiPackage).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_pi\_package\_revision\_date***

Запит дати ревізії пакета ПІ.

Об'явлення:

```
const char* gohub_pi_package_revision_date(  
    GohubPiPackage* piPackage);
```

Параметри:

`piPackage`

Об'єкт пакета ПІ (`GohubPiPackage`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_pi\_package\_revision\_date\_w***

Запит дати ревізії пакета ПІ (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pi_package_revision_date_w(  
    GohubPiPackage* piPackage);
```

Параметри:

`piPackage`

Об'єкт пакета ПІ (`GohubPiPackage`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_pi\_package\_status***

Запит статусу електронних даних ПІ.

Об'явлення:

```
int gohub_pi_package_status(  
    GohubPiPackage* piPackage);
```

Параметри:

`eData`

Об'єкт електронних даних ПІ (`GohubEData`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_pi\_package\_consignment\_id***

Запитати ID перевізного документа на який посилається пакет ПІ.

Об'явлення:

```
const char* gohub_pi_package_consignment_id(  
    GohubPiPackage* piPackage);
```

Параметри:

piPackage

Об'єкт пакета ПІ (GohubPiPackage).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_pi\_package\_consignment\_id\_w***

Запитати ID перевізного документа на який посилається пакет ПІ (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pi_package_consignment_id_w(  
    GohubPiPackage* piPackage);
```

Параметри:

piPackage

Об'єкт пакета ПІ (GohubPiPackage).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_pi\_package\_pipacktoed\_count***

Запитати кількість елементів GohubPiPackageToEData що містяться у пакеті ПІ.

Об'явлення:

```
int gohub_pi_package_pipacktoed_count(  
    GohubPiPackage* piPackage);
```

Параметри:

piPackage

Об'єкт пакета ПІ (GohubPiPackage).

Результат:

кількість елементів GohubPiPackageToEData що містяться у пакеті ПІ.

### ***gohub\_pi\_package\_pipacktoed***

Запит елементу GohubPiPackageToEData у пакеті ПІ за індексом.

Об'явлення:

```
GohubPiPackageToEData* gohub_pi_package_pipacktoed(  
    GohubPiPackage* piPackage,  
    int index);
```

Параметри:

piPackage

Об'єкт пакета ПІ (GohubPiPackage).

index

Номер індексу. Може мати значення від 0 до gohub\_pi\_package\_pipacktoed\_count()-1.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_pipacktoed\_id***

Запитати ID про'єкту `GohubPiPackageToEData`.

Об'явлення:

```
const char* gohub_pipacktoed_id(  
    GohubPiPackageToEData* piPackageToEData);
```

Параметри:

`piPackageToEData`

Об'єкт `GohubPiPackageToEData`.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_pipacktoed\_id\_w***

Запитати ID про'єкту `GohubPiPackageToEData` (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pipacktoed_id_w(  
    GohubPiPackageToEData* piPackageToEData);
```

Параметри:

`piPackageToEData`

Об'єкт `GohubPiPackageToEData`.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_pipacktoed\_edata\_id***

Запитати ID електронних даних в про'єкті `GohubPiPackageToEData`.

Об'явлення:

```
const char* gohub_pipacktoed_edata_id(  
    GohubPiPackageToEData* piPackageToEData);
```

Параметри:

`piPackageToEData`

Об'єкт `GohubPiPackageToEData`.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_pipacktoed\_edata\_id\_w***

Запитати ID електронних даних в про'єкті `GohubPiPackageToEData` (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pipacktoed_edata_id_w(  

```



```
GohubPiPackageToEData* piPackageToEData);
```

Параметри:

```
piPackageToEData
```

Об'єкт GohubPiPackageToEData.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_pipacktoed\_pi\_package\_id***

Запитати ID пакета ПІ в проєкті GohubPiPackageToEData.

Об'явлення:

```
const char* gohub_pipacktoed_pi_package_id(  
GohubPiPackageToEData* piPackageToEData);
```

Параметри:

```
piPackageToEData
```

Об'єкт GohubPiPackageToEData.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_pipacktoed\_pi\_package\_id\_w***

Запитати ID пакета ПІ в проєкті GohubPiPackageToEData (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pipacktoed_pi_package_id_w(  
GohubPiPackageToEData* piPackageToEData);
```

Параметри:

```
piPackageToEData
```

Об'єкт GohubPiPackageToEData.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_pipacktoed\_note***

Запитати примітки в проєкті GohubPiPackageToEData.

Об'явлення:

```
const char* gohub_pipacktoed_note(  
GohubPiPackageToEData* piPackageToEData);
```

Параметри:

```
piPackageToEData
```

Об'єкт GohubPiPackageToEData.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_pipacktoed\_note\_w***

Запитати Примітки в про'єкті GohubPiPackageToEData (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pipacktoed_note_w(  
    GohubPiPackageToEData* piPackageToEData);
```

Параметри:

piPackageToEData

Об'єкт GohubPiPackageToEData.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_pipacktoed\_edata\_version***

Запитати версію електронних даних в про'єкті GohubPiPackageToEData.

Об'явлення:

```
const char* gohub_pipacktoed_edata_version(  
    GohubPiPackageToEData* piPackageToEData);
```

Параметри:

piPackageToEData

Об'єкт GohubPiPackageToEData.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_pipacktoed\_edata\_version\_w***

Запитати версію електронних даних в про'єкті GohubPiPackageToEData (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_pipacktoed_edata_version_w(  
    GohubPiPackageToEData* piPackageToEData);
```

Параметри:

piPackageToEData

Об'єкт GohubPiPackageToEData.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_add\_edata\_to\_pi\_package***

Додати електронні дані до пакету ПІ.

Об'явлення:

```
GohubPiPackage* gohub_add_edata_to_pi_package(  
    GohubConnection* connection,  
    GohubEData* eData,  
    const char* piPackageId);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

eData

Об'єкт електронних даних ПІ (GohubEData).

piPackageId

Унікальний ідентифікатор електронних даних ПІ.

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_add\_edata\_to\_pi\_package\_w***

Додати електронні дані до пакету ПІ (у кодуванні UTF-16).

#### Об'явлення:

```
GohubPiPackage* gohub_add_edata_to_pi_package_w(  
    GohubConnection* connection,  
    GohubEData* eData,  
    const GohubWChar* piPackageId);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

eData

Об'єкт електронних даних ПІ (GohubEData).

piPackageId

Унікальний ідентифікатор супровідного документа (у кодуванні UTF-16).

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## **4.4.6. Робота з накопичувальними картками ФДУ-92**

gohub_query_fdu92	Запитати накопичувальну картку
gohub_query_fdu92_w	Запитати накопичувальну картку (в UTF-16)
gohub_query_fdu92_by_number	Запитати накопичувальну картку за номером станції та номером накопичувальної картки
gohub_query_fdu92_by_number_w	Запитати накопичувальну картку за номером станції та номером накопичувальної картки (в UTF-16)
gohub_query_next_fdu92	Запитати наступну накопичувальну картку
gohub_create_fdu92	Створити накопичувальну картку
gohub_create_fdu92_w	Створити накопичувальну картку (в UTF-16)

gohub_load_fdu92	Завантажити накопичувальну картку з файлу xml
gohub_load_fdu92_w	Завантажити накопичувальну картку з файлу xml (UTF-16)
gohub_save_fdu92	Зберегти накопичувальну картку
gohub_save_fdu92_w	Зберегти накопичувальну картку (в UTF-16)
gohub_send_fdu92	Відправити накопичувальну картку
gohub_fdu92_id	Запитати ID накопичувальної картки
gohub_fdu92_id_w	Запитати ID накопичувальної картки (в UTF-16)
gohub_fdu92_status	Запитати статус накопичувальної картки
fdu92_revision	Запитати номер ревізії накопичувальної картки
gohub_fdu92_text	Запитати текст накопичувальної картки
gohub_fdu92_text_w	Запитати текст накопичувальної картки (в UTF-16)
fdu92_size	Запитати передбачуваний розмір накопичувальної картки представлений в поточній кодової сторінці
gohub_fdu92_signer_info	Запитати інформацію про підписантів
gohub_fdu92_signer_info_w	Запитати інформацію про підписантів (в UTF-16)
gohub_fdu92_sign_time	Запитати інформацію про час підписання
gohub_fdu92_sign_time_w	Запитати інформацію про час підписання (в UTF-16)
gohub_fdu92_signer_name_w	Запитати ім'я підписанта (в UTF-16)
gohub_fdu92_has_signature	Перевірити чи підписаний документ
gohub_save_fdu92	Зберегти накопичувальну картку в файл
gohub_save_fdu92_w	Зберегти накопичувальну картку в файл (в UTF-16)
gohub_reject_fdu92	Відкликати накопичувальну картку
gohub_reject_fdu92_w	Відкликати накопичувальну картку (в UTF-16)
gohub_close_fdu92	Закрити накопичувальну картку

## *gohub\_query\_fdu92*

Запитати накопичувальну картку ФДУ-92.

Об'явлення:

```
GohubFdu92* gohub_query_fdu92(
GohubConnection* connection,
const char* fdu92Id);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

fdu92Id

Унікальний ідентифікатор накопичувальної картки.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## *gohub\_query\_fdu92\_w*

Запитати накопичувальну картку ФДУ-92.

Об'явлення:

```
GohubFdu92* gohub_query_fdu92_w(
GohubConnection* connection,
const GohubWChar* fdu92Id);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

fdu92Id

Унікальний ідентифікатор накопичувальної картки (у кодуванні UTF-16).

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_fdu92\_by\_number***

Запитати накопичувальну картку за номером станції та номером накопичувальної картки

#### Об'явлення:

```
GohubFdu92* gohub_query_fdu92_number(  
GohubConnection* connection,  
const char* registration_esr, const char* registration_num);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

registration\_esr

Номер станції.

registration\_num

Номер накопичувальної картки.

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_fdu92\_by\_number\_w***

Запитати накопичувальну картку за номером станції та номером накопичувальної картки (в UTF-16)

#### Об'явлення:

```
GohubFdu92* gohub_query_fdu92_w(  
GohubConnection* connection,  
const GohubWChar* registration_esr,  
const GohubWChar* registration_num);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

registration\_esr

Номер станції. (у кодуванні UTF-16).

registration\_num

Номер накопичувальної картки. (у кодуванні UTF-16).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_create\_fdu92***

Створити накопичувальну картку ФДУ-92.

Об'явлення:

```
GohubFdu92* gohub_create_fdu92(  
    const char* content);
```

Параметри:

content

Рядок з xml-структурою з вмістом документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_create\_fdu92\_w***

Створити накопичувальну картку ФДУ-92.

Об'явлення:

```
GohubFdu92* gohub_create_fdu92_w(  
    const GohubWChar* content);
```

Параметри:

content

Рядок з xml-структурою з вмістом документа у кодуванні UTF-16;.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_load\_fdu92***

Завантажити накопичувальну картку ФДУ-92 з файлу xml.

Об'явлення:

```
GohubFdu92* gohub_load_fdu92(  
    const char* id,  
    const char* path);
```

Параметри:

id

Ідентифікатор картки ФДУ-92 на сервері.

path

Шлях до файлу.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_load\_fdu92\_w***

Завантажити накопичувальну картку ФДУ-92 з файлу xml.

Об'явлення:

```
GohubFdu92* gohub_load_fdu92_w(  
const GohubWChar* id,  
const GohubWChar* path);
```

Параметри:

`id`

Ідентифікатор картки ФДУ-92 на сервері.

`path`

Шлях до файлу у кодуванні UTF-16.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_send\_fdu92***

Відправити накопичувальну картку ФДУ-92.

Об'явлення:

```
GohubBool gohub_send_fdu92(  
GohubConnection* connection,  
GohubFdu92* fdu92);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`

`fdu92`

Об'єкт документа (`GohubFdu92`) .;

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_query\_next\_fdu92***

Запитати наступну накопичувальну картку ФДУ-92.

Об'явлення:

```
GohubFdu92* gohub_query_next_fdu92(  
GohubConnection* connection,  
int lastRevision);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

lastRevision

Ревізія документа, від якої починається пошук наступної накопичувальної картки.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_fdu92\_id***

Запит ID накопичувальної картки ФДУ-92.

Об'явлення:

```
const char* gohub_fdu92_id(  
GohubFdu92* fdu92);
```

Параметри:

`fdu92`

Об'єкт документа (`GohubFdu92`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_fdu92\_id\_w***

Запит ID накопичувальної картки ФДУ-92 (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_fdu92_id_w(  
GohubFdu92* fdu92);
```

Параметри:

`fdu92`

Об'єкт документа (`GohubFdu92`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_fdu92\_status***

Запит статусу накопичувальної картки ФДУ-92.

Об'явлення:

```
GohubFdu92Status gohub_fdu92_status(  
GohubFdu92* fdu92);
```

Параметри:

`fdu92`

Об'єкт документа (`GohubFdu92`) , статус якого запитується;

Результат:

У разі успіху - значення з перерахування `GohubFdu92Status` . У разі помилки - «-1». Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .



## ***gohub\_fdu92\_revision***

Запит номера ревізії накопичувальної картки ФДУ-92.

Об'явлення:

```
int gohub_fdu92_revision(  
GohubFdu92* fdu92);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

fdu92

Об'єкт документа (GohubFdu92).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_fdu92\_text***

Запит тексту накопичувальної картки ФДУ-92.

Об'явлення:

```
const char* gohub_fdu92_text(  
GohubFdu92* fdu92);
```

Параметри:

fdu92

Об'єкт документа (GohubFdu92).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_fdu92\_text\_w***

Запит тексту накопичувальної картки ФДУ-92 (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_fdu92_text_w(  
GohubFdu92* fdu92);
```

Параметри:

fdu92

Об'єкт документа (GohubFdu92).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_fdu92\_size***

Запит передбачуваного розміру файлу, який вийде при збереженні документа у поточній кодуванні (GohubFdu92).

Об'явлення:

```
int gohub_fdu92_size(  
GohubFdu92* fdu92);
```

Параметри:

fdu92

Об'єкт документа (GohubFdu92) , текст якого запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_fdu92\_signer\_info***

Запит інформації про підписанта накопичувальної картки ФДУ-92.

Об'явлення:

```
const char* gohub_fdu92_signer_info(  
GohubFdu92* fdu92);
```

Параметри:

fdu92

Об'єкт документа (GohubFdu92) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_fdu92\_signer\_info\_w***

Запит інформації про підписанта накопичувальної картки ФДУ-92 (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_fdu92_signer_info_w(  
GohubFdu92* fdu92);
```

Параметри:

fdu92

Об'єкт документа (GohubFdu92) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_fdu92\_sign\_time***

Запит інформації про підписанта накопичувальної картки ФДУ-92.

Об'явлення:

```
const char* gohub_fdu92_sign_time(  
GohubFdu92* fdu92);
```

Параметри:

fdu92

Об'єкт документа (GohubFdu92) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_fdu92\_sign\_time\_w***

Запит інформації про підписанта накопичувальної картки ФДУ-92 (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_fdu92_sign_time_w(  
GohubFdu92* fdu92);
```

Параметри:

`fdu92`

Об'єкт документа (`GohubFdu92`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_fdu92\_sign\_name\_w***

Запит імені підписанта накопичувальної картки ФДУ-92 (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_fdu92_sign_name_w(  
GohubFdu92* fdu92);
```

Параметри:

`fdu92`

Об'єкт документа (`GohubFdu92`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_fdu92\_has\_signature***

Перевірити чи підписаний документ.

Об'явлення:

```
const GohubWChar* gohub_fdu92_has_signature(  
GohubFdu92* fdu92);
```

Параметри:

`fdu92`

Об'єкт документа (`GohubFdu92`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_save\_fdu92***

Збереження накопичувальної картки ФДУ-92.

Об'явлення:

```
GohubBool gohub_save_fdu92 (
GohubFdu92* fdu92,
const char* path,
int codePage);
```

**Параметри:**

fdu92

Об'єкт документа (GohubFdu92) , що зберігається на диск;

path

Шлях, куди зберігається файл;

codePage

Числове позначення сторінки кодування, в якій зберігається файл.

**Результат:**

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_save\_fdu92\_w***

Збереження накопичувальної картки ФДУ-92.

**Об'явлення:**

```
GohubBool gohub_save_fdu92_w (
GohubFdu92* fdu92,
const GohubWChar* path,
int codePage);
```

**Параметри:**

fdu92

Об'єкт документа (GohubFdu92) , що зберігається на диск;

path

Шлях, куди зберігається файл (у кодуванні UTF-16);

codePage

Числове позначення сторінки кодування, в якій зберігається файл.

**Результат:**

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_reject\_fdu92***

Відкликати накопичувальну картку ФДУ-92.

**Об'явлення:**

```
GohubBool gohub_reject_fdu92 (
GohubConnection* connection,
const char* fdu92Id);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`fdu92Id`

Унікальний ідентифікатор накопичувальної картки .

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_reject\_fdu92\_w***

Відкликати накопичувальну картку ФДУ-92.

Об'явлення:

```
GohubBool gohub_reject_fdu92_w(  
    GohubConnection* connection,  
    const GohubWChar* fdu92Id);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`fdu92Id`

Унікальний ідентифікатор накопичувальної картки (у кодуванні UTF-16).

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_close\_fdu92***

Закриття документа (`GohubFdu92`) .

Об'явлення:

```
GohubBool gohub_close_fdu92(  
    GohubFdu92* fdu92);
```

Параметри:

`fdu92`

Об'єкт документа (`GohubFdu92`) .

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

#### **4.4.7. Робота з відомостями про користування вагонами / контейнерами ГУ-46**

<code>gohub_query_gu46</code>	Запитати відомість про користування вагонами/контейнерами
<code>gohub_query_gu46_w</code>	Запитати відомість про користування вагонами/контейнерами (в UTF-16)
<code>gohub_query_next_gu46</code>	Запитати наступну відомість про користування вагонами/контейнерами
<code>gohub_create_gu46</code>	Створити відомість про користування

	вагонами/контейнерами
gohub_create_gu46_w	Створити відомість про користування вагонами/контейнерами
gohub_load_gu46	Завантажити відомість про користування вагонами/контейнерами з файлу xml
gohub_load_gu46_w	Завантажити відомість про користування вагонами/контейнерами з файлу xml (UTF-16)
gohub_send_gu46	Відправити відомість про користування вагонами/контейнерами
gohub_gu46_id	Запитати ID відомості про користування вагонами/контейнерами
gohub_gu46_id_w	Запитати ID відомості про користування вагонами/контейнерами (в UTF-16)
gohub_gu46_status	Запитати статус відомості про користування вагонами/контейнерами
gu46_revision	Запитати номер ревізії відомості про користування вагонами/контейнерами
gohub_gu46_text	Запитати текст відомості про користування вагонами/контейнерами
gohub_gu46_text_w	Запитати текст відомості про користування вагонами/контейнерами (в UTF-16)
gu46_size	Запитати передбачуваний розмір відомості про користування вагонами/контейнерами представлений в поточній кодової сторінці
gohub_gu46_signer_info	Запитати інформацію про підписантів відомості про користування вагонами/контейнерами
gohub_gu46_signer_info_w	Запитати інформацію про підписантів відомості про користування вагонами/контейнерами (в UTF-16)
gohub_gu46_sign_time	Запитати час підписання відомості про користування вагонами/контейнерами
gohub_gu46_sign_time_w	Запитати час підписання відомості про користування вагонами/контейнерами (в UTF-16)
gohub_gu46_signer_name_w	Запитати ім'я підписанта (в UTF-16)
gohub_gu46_has_signature	Перевірити чи підписаний документ
gohub_save_gu46	Зберегти відомість про користування вагонами/контейнерами
gohub_save_gu46_w	Зберегти відомість про користування вагонами/контейнерами (в UTF-16)
gohub_reject_gu46	Відкликати відомість про користування вагонами/контейнерами
gohub_reject_gu46_w	Відкликати відомість про користування вагонами/контейнерами (в UTF-16)
gohub_close_gu46	Закрити відомість про користування вагонами/контейнерами

### ***gohub\_query\_gu46***

Запитати відомість про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubGu46* gohub_query_gu46(  
GohubConnection* connection,  
const char* gu46Id);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu46Id

Унікальний ідентифікатор відомості про користування вагонами/контейнерами.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_query\_gu46\_w***

Запитати відомість про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubGu46* gohub_query_gu46_w(  
GohubConnection* connection,  
const GohubWChar* gu46Id);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu46Id

Унікальний ідентифікатор відомості про користування вагонами/контейнерами (у кодуванні UTF-16).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_query\_next\_gu46***

Запитати наступну відомість про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubGu46* gohub_query_next_gu46(  
GohubConnection* connection,  
int lastRevision);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

lastRevision

Ревізія документа, від якої починається пошук наступної відомості про користування вагонами/контейнерами.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_create\_gu46***

Створити відомість про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubGu46* gohub_create_gu46(  
const char* gu46Id,  
const char* content);
```

Параметри:

`gu46Id`

Унікальний ідентифікатор відомості про користування вагонами/контейнерами;

`content`

Рядок з xml-структурою з вмістом документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_create\_gu46\_w***

Створити відомість про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubGu46* gohub_create_gu46_w(  
const GohubWChar* gu46Id,  
const GohubWChar* content);
```

Параметри:

`gu46Id`

Унікальний ідентифікатор відомості про користування вагонами/контейнерами (у кодуванні UTF-16);

`content`

Рядок з xml-структурою з вмістом документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_load\_gu46***

Завантажити відомість про користування вагонами/контейнерами ГУ-46 з файлу xml.

Об'явлення:



```
GohubGu46* gohub_load_gu46(  
const char* id,  
const char* path);
```

**Параметри:**

id

Ідентифікатор відомості ГУ-46 на сервері.

path

Шлях до файлу.

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_load\_gu46\_w***

Завантажити відомість про користування вагонами/контейнерами ГУ-46 з файлу xml.

**Об'явлення:**

```
GohubGu46* gohub_load_gu46_w(  
const GohubWChar* id,  
const GohubWChar* path);
```

**Параметри:**

id

Ідентифікатор відомості ГУ-46 на сервері.

path

Шлях до файлу у кодуванні UTF-16.

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_send\_gu46***

Відправити відомість про користування вагонами/контейнерами ГУ-46.

**Об'явлення:**

```
GohubBool gohub_send_gu46(  
GohubConnection* connection,  
GohubGu46* gu46);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu46

Об'єкт документа (GohubGu46).

**Результат:**

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_gu46\_id***

Запит ID відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const char* gohub_gu46_id(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_id\_w***

Запит ID відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const GohubWChar* gohub_gu46_id_w(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_status***

Запит статусу відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubGu46Status gohub_gu46_status(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46) , статус якого запитується;

Результат:

У разі успіху - значення з перерахування GohubGu46Status . У разі помилки - «-1». Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_revision***

Запит номера ревізії відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
int gohub_gu46_revision(  
GohubGu46* gu46);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_text***

Запит тексту відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const char* gohub_gu46_text(  
GohubGu46* gu46) ;
```

Параметри:

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_text\_w***

Запит тексту відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const GohubWChar* gohub_gu46_text_w(  
GohubGu46* gu46) ;
```

Параметри:

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_size***

Запит передбачуваного розміру файлу, який вийде при збереженні документа у поточній кодуванні (GohubGu46) .

Об'явлення:

```
int gohub_gu46_size(  
GohubGu46* gu46) ;
```

Параметри:

gu46

Об'єкт документа (GohubGu46) , текст якого запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_gu46\_signer\_info***

Запит інформації про підписанта відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const char* gohub_gu46_signer_info(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_gu46\_signer\_info\_w***

Запит інформації про підписанта відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const GohubWChar* gohub_gu46_signer_info_w(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_gu46\_sign\_time***

Запит часу підписання відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const char* gohub_gu46_sign_time(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_gu46\_sign\_time\_w***

Запит часу підписання відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
const GohubWChar* gohub_gu46_sign_time_w(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_sign\_name\_w***

Запит імені підписанта відомості про користування вагонами/контейнерами ГУ-46 (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_gu46_sign_name_w(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu46\_has\_signature***

Перевірити чи підписаний документ.

Об'явлення:

```
const GohubWChar* gohub_gu46_has_signature(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_save\_gu46***

Збереження відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubBool gohub_save_gu46(  
GohubGu46* gu46,  
const char* path,  
int codePage);
```

Параметри:

gu46

Об'єкт документа (GohubGu46) , що зберігається на диск;

path

шлях, куди зберігається файл;

codePage

числове позначення сторінки кодування, в якій зберігається файл.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_save\_gu46\_w***

Збереження відомості про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubBool gohub_save_gu46_w(  
GohubGu46* gu46,  
const GohubWChar* path,  
int codePage);
```

Параметри:

gu46

Об'єкт документа (`GohubGu46`), що зберігається на диск;

path

Шлях, куди зберігається файл (у кодуванні UTF-16);

codePage

Числове позначення сторінки кодування, в якій зберігається файл.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_reject\_gu46***

Відкликати відомість про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubBool gohub_reject_gu46(  
GohubConnection* connection,  
const char* gu46Id);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

gu46Id

Унікальний ідентифікатор відомості про користування вагонами/контейнерами.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## *gohub\_reject\_gu46\_w*

Відкликати відомість про користування вагонами/контейнерами ГУ-46.

Об'явлення:

```
GohubBool gohub_reject_gu46_w(  
GohubConnection* connection,  
const GohubWChar* gu46Id);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu46Id

Унікальний ідентифікатор відомості про користування вагонами/контейнерами (у кодуванні UTF-16).

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## *gohub\_close\_gu46*

Закриття документа (GohubGu46).

Об'явлення:

```
GohubBool gohub_close_gu46(  
GohubGu46* gu46);
```

Параметри:

gu46

Об'єкт документа (GohubGu46).

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### **4.4.8. Робота з пам'ятками про подачу / прибирання вагонів та видачу / прийом контейнерів ГУ-45**

gohub_query_gu45	Запитати пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів
gohub_query_gu45_w	Запитати пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів (в UTF-16)
gohub_query_next_gu45	Запитати наступну пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів
gohub_gu45_id	Запитати ID пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів
gohub_gu45_id_w	Запитати ID пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів (в UTF-16)
gohub_gu45_status	Запитати статус пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів
gu45_revision	Запитати номер ревізії пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів
gohub_gu45_text	Запитати текст пам'ятки про подачу/прибирання вагонів та

	видачу/прийом контейнерів
gohub_gu45_text_w	Запитати текст пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів (в UTF-16)
gohub_gu45_signer_info	Запитати інформацію про підписантів пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів
gohub_gu45_signer_info_w	Запитати інформацію про підписантів пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів (в UTF-16)
gohub_gu45_sign_time	Запитати час підписання пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів
gohub_gu45_sign_time_w	Запитати час підписання пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів (в UTF-16)
gohub_gu45_signer_name_w	Запитати ім'я підписанта (в UTF-16)
gohub_gu45_has_signature	Перевірити чи підписаний документ
gu45_size	Запитати передбачуваний розмір пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів представлений в поточній кодової сторінці
gohub_save_gu45	Зберегти пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів
gohub_save_gu45_w	Зберегти пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів (в UTF-16)
gohub_close_gu45	Закрити пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів

## ***gohub\_query\_gu45***

Запитати пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
GohubGu45* gohub_query_gu45 (
GohubConnection* connection,
const char* gu45Id);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu45Id

Унікальний ідентифікатор пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_gu45\_w***

Запитати пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:



```
GohubGu45* gohub_query_gu45_w(  
GohubConnection* connection,  
const GohubWChar* gu45Id);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu45Id

Унікальний ідентифікатор пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів (у кодуванні UTF-16).

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_query\_next\_gu45***

Запитати наступну пам'ятку про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

**Об'явлення:**

```
GohubGu45* gohub_query_next_gu45(  
GohubConnection* connection,  
int lastRevision);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

lastRevision

Ревізія документа, від якої починається пошук наступної пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів.

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_gu45\_id***

Запит ID пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

**Об'явлення:**

```
const char* gohub_gu45_id(  
GohubGu45* gu45);
```

**Параметри:**

gu45

Об'єкт документа (GohubGu45) .

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_gu45\_id\_w***

Запит ID пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
const GohubWChar* gohub_gu45_id_w(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_gu45\_status***

Запит статусу пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
GohubGu45Status gohub_gu45_status(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45) , статус якого запитується;

Результат:

У разі успіху - значення з перерахування GohubGu45Status . У разі помилки - «-1». Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_gu45\_revision***

Запит номера ревізії пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
int gohub_gu45_revision(  
GohubGu45* gu45);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

gu45

Об'єкт документа (GohubGu45) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_gu45\_text***

Запит тексту пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
const char* gohub_gu45_text(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_gu45\_text\_w***

Запит тексту пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
const GohubWChar* gohub_gu45_text_w(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_gu45\_size***

Запит передбачуваного розміру файлу, який вийде при збереженні документа у поточній кодуванні (GohubGu45).

Об'явлення:

```
int gohub_gu45_size(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45), текст якого запитується;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_gu45\_signer\_info***

Запит інформації про пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
const char* gohub_gu45_signer_info(  
GohubGu46* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu46).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_gu45\_signer\_info\_w***

Запит інформації про пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
const GohubWChar* gohub_gu45_signer_info_w(  
GohubGu46* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu46).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_gu45\_sign\_time***

Запит часу підписання пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
const char* gohub_gu45_sign_time(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu46).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_gu45\_sign\_time\_w***

Запит часу підписання пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
const GohubWChar* gohub_gu45_sign_time_w(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_gu45\_sign\_name\_w***

Запит імені підписанта пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45 (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_gu45_sign_name_w(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_gu45\_has\_signature***

Перевірити чи підписаний документ.

Об'явлення:

```
const GohubWChar* gohub_gu45_has_signature(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (GohubGu45) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_save\_gu45***

Збереження пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
GohubBool gohub_save_gu45(  
GohubGu45* gu45,  
const char* path,  
int codePage);
```

Параметри:

gu45

Об'єкт документа (GohubGu45) , що зберігається на диск;

path

Шлях, куди зберігається файл;

codePage

Числове позначення сторінки кодування, в якій зберігається файл.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_save\_gu45\_w***

Збереження пам'ятки про подачу/прибирання вагонів та видачу/прийом контейнерів ГУ-45.

Об'явлення:

```
GohubBool gohub_save_gu45_w(  
GohubGu45* gu45,  
const GohubWChar* path,  
int codePage);
```

Параметри:

gu45

Об'єкт документа (`GohubGu45`) , що зберігається на диск;

path

Шлях, куди зберігається файл (у кодуванні UTF-16);

codePage

Числове позначення сторінки кодування, в якій зберігається файл.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_close\_gu45***

Закриття документа (`GohubGu45`) .

Об'явлення:

```
GohubBool gohub_close_gu45(  
GohubGu45* gu45);
```

Параметри:

gu45

Об'єкт документа (`GohubGu45`) .

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

#### **4.4.9. Отримання друкованої форми документ**

<code>gohub_query_and_save_document_printable_form</code>	Запитати друковану форму документа
<code>gohub_query_and_save_document_printable_form_w</code>	Запитати друковану форму документа (в UTF-16)

## ***gohub\_query\_and\_save\_document\_printable\_form***

Запитати друковану форму документа за його ID та зберегти її в файл.

Об'явлення:

```
GohubBool gohub_query_and_save_document_printable_form(  
    GohubConnection* connection,  
    const char* documentId,  
    const char* path);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentId

Унікальний ідентифікатор документа;

path

Шлях, за яким зберегти запитану друковану форму.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_and\_save\_document\_printable\_form\_w***

Запитати друковану форму документа за його ID (у кодуванні UTF-16) та зберегти її в файл.

Об'явлення:

```
GohubBool gohub_query_and_save_document_printable_form_w(  
    GohubConnection* connection,  
    const GohubWChar* documentId,  
    const GohubWChar* path);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentId

Унікальний ідентифікатор документа (у кодуванні UTF-16);

path

Шлях, за яким зберегти запитану друковану форму.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

#### 4.4.10. Робота з фільтрами для запитуваних документів

Нижче перераховані функції встановлюють фільтри, що впливають на функції запиту документів з п. 4.4.3.

gohub_clear_all_filters	Скинути всі фільтри
gohub_set_filter_by_document_status	Встановити фільтр за статусом документа
gohub_set_filter_by_document_number	Встановити фільтр за номером документа
gohub_set_filter_by_document_number_w	Встановити фільтр за номером документа (в UTF-16)
gohub_set_filter_by_wagon_number	Встановити фільтр за номером вагона
gohub_set_filter_by_wagon_number_w	Встановити фільтр за номером вагона (в UTF-16)
gohub_set_filter_by_departure_client	Встановити фільтр за кодом відправника
gohub_set_filter_by_departure_client_w	Встановити фільтр за кодом відправника (в UTF-16)
gohub_set_filter_by_departure_payer	Встановити фільтр за кодом платника по відправленню
gohub_set_filter_by_departure_payer_w	Встановити фільтр за кодом платника по відправленню (в UTF-16)
gohub_set_filter_by_departure_station	Встановити фільтр за кодом станції відправлення
gohub_set_filter_by_departure_station_w	Встановити фільтр за кодом станції відправлення (в UTF-16)
gohub_set_filter_by_arrival_client	Встановити фільтр за кодом одержувача
gohub_set_filter_by_arrival_client_w	Встановити фільтр за кодом одержувача (в UTF-16)
gohub_set_filter_by_arrival_payer	Встановити фільтр за кодом платника по прибуттю
gohub_set_filter_by_arrival_payer_w	Встановити фільтр за кодом платника по прибуттю (в UTF-16)
gohub_set_filter_by_arrival_station	Встановити фільтр за кодом станції призначення
gohub_set_filter_by_arrival_station_w	Встановити фільтр за кодом станції призначення (в UTF-16)
gohub_get_filter_by_document_status	Запитати значення фільтру за статусом документа
gohub_get_filter_by_document_number	Запитати значення фільтру за номером документа
gohub_get_filter_by_document_number_w	Запитати значення фільтру за номером документа (в UTF-16)
gohub_get_filter_by_wagon_number	Запитати значення фільтру за номером вагона
gohub_get_filter_by_wagon_number_w	Запитати значення фільтру за номером вагона (в UTF-16)
gohub_get_filter_by_departure_client	Запитати значення фільтру за кодом відправника
gohub_get_filter_by_departure_client_w	Запитати значення фільтру за кодом відправника (в UTF-16)
gohub_get_filter_by_departure_payer	Запитати значення фільтру за кодом платника по відправленню
gohub_get_filter_by_departure_payer_w	Запитати значення фільтру за кодом платника по відправленню (в UTF-16)
gohub_get_filter_by_departure_station	Запитати значення фільтру за кодом станції відправлення
gohub_get_filter_by_departure_station_w	Запитати значення фільтру за кодом станції відправлення (в UTF-16)
gohub_get_filter_by_arrival_client	Запитати значення фільтру за кодом одержувача
gohub_get_filter_by_arrival_client_w	Запитати значення фільтру за кодом одержувача (в UTF-16)
gohub_get_filter_by_arrival_payer	Запитати значення фільтру за кодом платника по прибуттю
gohub_get_filter_by_arrival_payer_w	Запитати значення фільтру за кодом платника по прибуттю (в UTF-16)
gohub_get_filter_by_arrival_station	Запитати значення фільтру за кодом станції призначення
gohub_get_filter_by_arrival_station_w	Запитати значення фільтру за кодом станції призначення (в UTF-16)



## ***gohub\_clear\_all\_filters***

Скинути всі налаштування фільтрів.

Об'явлення:

```
GohubBool gohub_clear_all_filters(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_set\_filter\_by\_document\_status***

Встановити фільтр за статусом документа.

Об'явлення:

```
GohubBool gohub_set_filter_by_document_status(  
    GohubConnection* connection,  
    int documentStatusCode);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentStatusCode

Код статусу документа, якщо вказати 0 – фільтр буде скинуто на «всі»;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_set\_filter\_by\_document\_number***

Встановити фільтр за номером документа.

Об'явлення:

```
GohubBool gohub_set_filter_by_document_number(  
    GohubConnection* connection,  
    const char* documentNumber);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

documentNumber

Номер документа, якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_document\_number\_w***

Встановити фільтр за номером документа (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_set_filter_by_document_number_w(  
    GohubConnection* connection,  
    const GohubWChar* documentNumber);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`documentNumber`

Номер документа (у кодуванні UTF-16), якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_wagon\_number***

Встановити фільтр за номером вагона.

Об'явлення:

```
GohubBool gohub_set_filter_by_wagon_number(  
    GohubConnection* connection,  
    const char* wagonNumber);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`wagonNumber`

Номер вагона, якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_wagon\_number\_w***

Встановити фільтр за номером вагона (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_set_filter_by_wagon_number_w(  
    GohubConnection* connection,  
    const GohubWChar* wagonNumber);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`wagonNumber`

Номер вагона (у кодуванні UTF-16), якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_departure\_client***

Встановити фільтр за кодом відправника.

Об'явлення:

```
GohubBool gohub_set_filter_by_departure_client(  
    GohubConnection* connection,  
    const char* clientCode);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`clientCode`

Код відправника, якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_departure\_client\_w***

Встановити фільтр за кодом відправника (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_set_filter_by_departure_client_w(  
    GohubConnection* connection,  
    const GohubWChar* clientCode);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`clientCode`

Код відправника (у кодуванні UTF-16), якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_set\_filter\_by\_departure\_payer***

Встановити фільтр за кодом платника по відправленню.

Об'явлення:

```
GohubBool gohub_set_filter_by_departure_payer(  
    GohubConnection* connection,  
    const char* payerCode);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

payerCode

Код платника по відправленню, якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_set\_filter\_by\_departure\_payer\_w***

Встановити фільтр за кодом платника по відправленню (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_set_filter_by_departure_payer_w(  
    GohubConnection* connection,  
    const GohubWChar* payerCode);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

payerCode

Код платника по відправленню (у кодуванні UTF-16), якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_set\_filter\_by\_departure\_station***

Встановити фільтр за кодом станції відправлення.

Об'явлення:

```
GohubBool gohub_set_filter_by_departure_station(  
    GohubConnection* connection,  
    const char* stationCode);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

stationCode

Код станції відправлення, якщо передати порожній покажчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_departure\_station\_w***

Встановити фільтр за кодом станції відправлення (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_set_filter_by_departure_station_w(  
    GohubConnection* connection,  
    const GohubWChar* stationCode);
```

Параметри:

connection

Покажчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

stationCode

Код станції відправлення (у кодуванні UTF-16), якщо передати порожній покажчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_arrival\_client***

Встановити фільтр за кодом одержувача.

Об'явлення:

```
GohubBool gohub_set_filter_by_arrival_client(  
    GohubConnection* connection,  
    const char* clientCode);
```

Параметри:

connection

Покажчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

clientCode

Код одержувача, якщо передати порожній покажчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_set\_filter\_by\_arrival\_client\_w***

Встановити фільтр за кодом одержувача (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_set_filter_by_arrival_client_w(  
    GohubConnection* connection,  
    const GohubWChar* clientCode);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

clientCode

Код одержувача (у кодуванні UTF-16), якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_set\_filter\_by\_arrival\_payer***

Встановити фільтр за кодом платника по прибуттю.

**Об'явлення:**

```
GohubBool gohub_set_filter_by_arrival_payer(  
    GohubConnection* connection,  
    const char* payerCode);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

payerCode

Код платника по прибуттю, якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_set\_filter\_by\_arrival\_payer\_w***

Встановити фільтр за кодом платника по прибуттю (у кодуванні UTF-16).

**Об'явлення:**

```
GohubBool gohub_set_filter_by_arrival_payer_w(  
    GohubConnection* connection,  
    const GohubWChar* payerCode);
```

**Параметри:**

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

payerCode

Код платника по прибуттю (у кодуванні UTF-16), якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_set\_filter\_by\_arrival\_station***

Встановити фільтр за кодом станції призначення.

Об'явлення:

```
GohubBool gohub_set_filter_by_arrival_station(  
    GohubConnection* connection,  
    const char* stationCode);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`stationCode`

Код станції призначення, якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_set\_filter\_by\_arrival\_station\_w***

Встановити фільтр за кодом станції призначення (у кодуванні UTF-16).

Об'явлення:

```
GohubBool gohub_set_filter_by_arrival_station_w(  
    GohubConnection* connection,  
    const GohubWChar* stationCode);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`stationCode`

Код станції призначення (у кодуванні UTF-16), якщо передати порожній показчик або порожній рядок – фільтр буде скинуто на «всі».

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_get\_filter\_by\_document\_status***

Запитати значення фільтру за статусом документа. Повертає значення цілого типу, що описано переліком `GohubDocumentStatus`.

Об'явлення:

```
GohubDocumentStatus gohub_get_filter_by_document_status(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_get\_filter\_by\_document\_number***

Запитати значення фільтру за номером документа.

Об'явлення:

```
const char* gohub_get_filter_by_document_number(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_get\_filter\_by\_document\_number\_w***

Запитати значення фільтру за номером документа (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_document_number_w(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_get\_filter\_by\_wagon\_number***

Запитати значення фільтру за номером вагона.

Об'явлення:

```
const char* gohub_get_filter_by_wagon_number(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.



Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_get\_filter\_by\_wagon\_number\_w***

Запитати значення фільтру за номером вагона (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_wagon_number_w(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_get\_filter\_by\_departure\_client***

Запитати значення фільтру за кодом відправника.

Об'явлення:

```
const char* gohub_get_filter_by_departure_client(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_get\_filter\_by\_departure\_client\_w***

Запитати значення фільтру за кодом відправника (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_departure_client_w(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_get\_filter\_by\_departure\_payer***

Запитати значення фільтру за кодом платника по відправленню.

Об'явлення:

```
const char* gohub_get_filter_by_departure_payer(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_get\_filter\_by\_departure\_payer\_w***

Запитати значення фільтру за кодом платника по відправленню (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_departure_payer_w(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_get\_filter\_by\_departure\_station***

Запитати значення фільтру за кодом станції відправлення.

Об'явлення:

```
const char* gohub_get_filter_by_departure_station(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_get\_filter\_by\_departure\_station\_w***

Запитати значення фільтру за кодом станції відправлення (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_departure_station_w(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_get\_filter\_by\_arrival\_client***

Запитати значення фільтру за кодом одержувача.

Об'явлення:

```
const char* gohub_get_filter_by_arrival_client(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_get\_filter\_by\_arrival\_client\_w***

Запитати значення фільтру за кодом одержувача (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_arrival_client_w(  
    GohubConnection* connection);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_get\_filter\_by\_arrival\_payer***

Запитати значення фільтру за кодом платника по прибуттю.

Об'явлення:

```
const char* gohub_get_filter_by_arrival_payer(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_get\_filter\_by\_arrival\_payer\_w***

Запитати значення фільтру за кодом платника по прибуттю (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_arrival_payer_w(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_get\_filter\_by\_arrival\_station***

Запитати значення фільтру за кодом станції прибуття.

Об'явлення:

```
const char* gohub_get_filter_by_arrival_station(  
    GohubConnection* connection);  
  
const char* gohub_get_filter_by_arrival_payer(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_get\_filter\_by\_arrival\_station\_w***

Запитати значення фільтру за кодом станції прибуття (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_filter_by_arrival_station_w(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect.

Результат:

У разі успіху - не порожній рядок. Якщо результат порожній рядок - значить фільтр не накладений або з'єднання не дійсне. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### **4.4.11.      Перевірка електронно-цифрового підпису**

gohub_document_has_signature	Дізнатися, чи містить документ електронно-цифровий підпис
gohub_document_check_signature	Виконати перевірку електронно-цифрового підпису документа
gohub_document_signer_name	Отримати ім'я особи, яка підписала електронний документ
gohub_document_signer_name_w	Отримати ім'я особи, яка підписала електронний документ (в UTF-16)
gohub_document_signer_info	Отримати детальну інформацію про особу, що підписала електронний документ
gohub_document_signer_info_w	Отримати детальну інформацію про особу, що підписала електронний документ (в UTF-16)
gohub_document_sign_time	Отримати позначку часу електронно-цифрового підпису
gohub_document_sign_time_w	Отримати позначку часу електронно-цифрового підпису (в UTF-16)

## ***gohub\_document\_has\_signature***

Перевірка, чи є у документа електронний підпис (GohubDocument).

Об'явлення:

```
GohubBool gohub_document_has_signature(  
  
    GohubDocument* document);
```

Параметри:

document

Об'єкт документа (GohubDocument);

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_document\_check\_signature***

Перевірка відповідності документа його електронному підпису.

Об'явлення:

```
GohubBool gohub_document_check_signature(  

```

```
GohubDocument* document);
```

**Параметри:**

document

Документ;

**Результат:**

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_signature\_name***

Запит імені особи, яка підписала документ.

**Об'явлення:**

```
const char* gohub_document_signer_name(  
  
GohubDocument* document);
```

**Параметри:**

document

Документ;

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_signature\_name\_w***

Запит імені особи, яка підписала документ, результат буде у кодуванні UTF-16.

**Об'явлення:**

```
const GohubWChar* gohub_document_signer_name_w(  
  
GohubDocument* document);
```

**Параметри:**

document

Документ;

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_signer\_info***

Запит інформації про підписанта документу – повертається форматований рядок з ключами полів та їх значеннями, через роздільник.

**Об'явлення:**

```
const char* gohub_document_signer_info(
```

```
GohubDocument* document);
```

Параметри:

document

Документ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_signer\_info\_w***

Запит інформації про підписанта документу – повертається форматований рядок (у кодуванні UTF-16) з ключами полів та їх значеннями, через роздільник.

Об'явлення:

```
const GohubWChar* gohub_document_signer_info_w(  
GohubDocument* document);
```

Параметри:

document

Документ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_sign\_time***

Запит інформації про час підписання документа.

Об'явлення:

```
const char* gohub_document_sign_time(  
GohubDocument* document);
```

Параметри:

document

Документ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_sign\_time\_w***

Запит інформації про час підписання документа, повертає значення у кодуванні UTF-16.

Об'явлення:

```
const GohubWChar* gohub_document_sign_time_w(  
GohubDocument* document);
```

Параметри:

document

Документ;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

#### 4.4.12. Накладення електронно-цифрового підпису

<code>gohub_open_private_key</code>	Відкрити електронний ключ за паролем
<code>gohub_open_private_key_w</code>	Відкрити електронний ключ за паролем (в UTF-16)
<code>gohub_open_private_key_by_bytes</code>	Відкрити електронний ключ у вигляді масиву байт за паролем
<code>gohub_open_private_key_by_bytes_w</code>	Відкрити електронний ключ у вигляді масиву байт за паролем (в UTF-16)
<code>gohub_open_private_key_from_path</code>	Відкрити електронний ключ за шляхом та паролем
<code>gohub_open_private_key_from_path_w</code>	Відкрити електронний ключ за шляхом та паролем (в UTF-16)
<code>gohub_private_key_owner_name</code>	Отримати ім'я власника електронного ключа
<code>gohub_private_key_owner_name_w</code>	Отримати ім'я власника електронного ключа (в UTF-16)
<code>gohub_private_key_owner_info</code>	Отримати детальну інформацію про власника електронного ключа
<code>gohub_private_key_owner_info_w</code>	Отримати детальну інформацію про власника електронного ключа (в UTF-16)
<code>gohub_sign_document</code>	Підписати документ електронно-цифровим підписом
<code>gohub_close_private_key</code>	Завершити роботу з електронним ключем

#### ***gohub\_open\_private\_key***

Відкриття сесії роботи електронно-цифрового ключа.

Об'явлення:

```
GohubBool gohub_open_private_key(  
    GohubConnection* connection,  
    const char* passwordToKey);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`passwordToKey`

Рядок з паролем від електронного ключа ;

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

#### ***gohub\_open\_private\_key\_w***

Відкриття сесії роботи електронно-цифрового ключа.

Об'явлення:

```
GohubBool gohub_open_private_key_w(  
    GohubConnection* connection,  
    const GohubWChar* passwordToKey);
```

Параметри:

`connection`



Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції  
`gohub_connect`;

`passwordToKey`

Рядок з паролем від електронного ключа у кодуванні UTF-16;

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_open\_private\_key\_by\_bytes***

Відкриття сесії роботи електронно-цифрового ключа, що надано масивом байт.

Об'явлення:

```
GohubBool gohub_open_private_key(  
    GohubConnection* connection,  
    const char* passwordToKey,  
    unsigned char* keyBinary,  
    unsigned int length);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції  
`gohub_connect`;

`passwordToKey`

Рядок з паролем від електронного ключа ;

`keyBinary`

Електронний ключ у вигляді масиву байт ;

`length`

Довжина масиву байт ;

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_open\_private\_key\_by\_bytes\_w***

Відкриття сесії роботи електронно-цифрового ключа, що надано масивом байт.

Об'явлення:

```
GohubBool gohub_open_private_key_w(  
    GohubConnection* connection,  
    const GohubWChar* passwordToKey,  
    unsigned char* keyBinary,  
    unsigned int length);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції  
`gohub_connect`;

passwordToKey

Рядок з паролем від електронного ключа у кодуванні UTF-16;

keyBinary

Електронний ключ у вигляді масиву байт;

length

Довжина масиву байт;

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_open\_private\_key\_from\_path***

Відкриття сесії роботи електронно-цифрового ключа, що розташований за вказаним шляхом.

Об'явлення:

```
GohubBool gohub_open_private_key(  
    GohubConnection* connection,  
    const char* passwordToKey,  
    const char* keyFileName);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

passwordToKey

Рядок з паролем від електронного ключа;

keyFileName

Шлях до електронного ключу;

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_open\_private\_key\_from\_path\_w***

Відкриття сесії роботи електронно-цифрового ключа, що розташований за вказаним шляхом.

Об'явлення:

```
GohubBool gohub_open_private_key_w(  
    GohubConnection* connection,  
    const GohubWChar* passwordToKey,  
    const GohubWChar* keyFileName);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

passwordToKey

Рядок з паролем від електронного ключа у кодуванні UTF-16;

keyFileName

Шлях до електронного ключу у кодуванні UTF-16;

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_private\_key\_owner\_name***

Повертає ім'я власника електронно-цифрового ключа (у кодуванні UTF-16), робоча сесія якого зараз відкрита.

Об'явлення:

```
const char* gohub_private_key_owner_name(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_private\_key\_owner\_name\_w***

Повертає ім'я власника електронно-цифрового ключа (у кодуванні UTF-16), робоча сесія якого зараз відкрита.

Об'явлення:

```
const GohubWChar* gohub_private_key_owner_name_w(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_private\_key\_owner\_info***

Запит інформації про власника електронно-цифрового ключа, робоча сесія якого зараз відкрита – повертається форматований рядок з ключами полів та їх значеннями, через роздільник.

Об'явлення:

```
const char* gohub_private_key_owner_info(  
  
    GohubConnection* connection);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_private\_key\_owner\_info\_w***

Запит інформації про власника електронно-цифрового ключа, робоча сесія якого зараз відкрита – повертається форматований рядок (у кодуванні UTF-16) з ключами полів та їх значеннями, через роздільник.

#### Об'явлення:

```
const GohubWChar* gohub_private_key_owner_info_w(  
  
    GohubConnection* connection);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

#### Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_sign\_document***

Підписання документа електронно-цифровим ключем, сесія якого зараз відкрита.

#### Об'явлення:

```
GohubBool gohub_sign_document(  
    GohubConnection* connection,  
    GohubDocument* document);
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

document

Документ;

#### Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_close\_private\_key***

Закриття робочої сесії електронно-цифрового ключа.

Об'явлення:

```
GohubBool gohub_close_private_key(  
    GohubConnection* connection);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

#### 4.4.13. Операції з файлами електронних ключів

Цей блок функцій дозволяє зробити доступним для використання файли електронних ключів, розміщених в довільному місці файлової системи. Монтювання файлу ключа по своєму ефекту аналогічно підключенню зовнішнього носія (наприклад, модуля флеш-пам'яті) з електронним ключем.

**ВАЖЛИВО!** Підключення електронного ключа розповсюджується тільки на поточного користувача.

gohub_mount_file_key	Монтювати електронний ключ з вказаного файлу
gohub_mount_file_key_w	Монтювати електронний ключ з вказаного файлу (в UTF-16)
gohub_unmount_file_key	Демонтювати електронний ключ з файлу
gohub_unmount_file_key_w	Демонтювати електронний ключ з файлу (в UTF-16)
gohub_query_mounted_file_keys	Запитати кількість монтюваних з файлів електронних ключів
gohub_mounted_file_key_id	Запитати ID електронного ключа за його порядковим номером у переліку
gohub_mounted_file_key_id_w	Запитати ID електронного ключа за його порядковим номером у переліку (в UTF-16)
gohub_mounted_file_key_dir	Запитати шлях до електронного ключу за його порядковим номером у переліку
gohub_mounted_file_key_dir_w	Запитати шлях до електронного ключу за його порядковим номером у переліку (в UTF-16)

#### *gohub\_mount\_file\_key*

Монтювати електронний ключ з вказаного файлу.

Об'явлення:

```
GohubBool gohub_mount_file_key(  
    const char* keyId,  
    const char* path);
```

Параметри:

keyId

Ідентифікатор ключа;

path

Шлях до файлу ключа.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_mount\_file\_key\_w***

Монтувати електронний ключ з вказаного файлу.

Об'явлення:

```
GohubBool gohub_mount_file_key_w(  
    const GohubWChar* keyId,  
    const GohubWChar* path);
```

Параметри:

keyId

Ідентифікатор ключа у кодуванні UTF-16;

path

Шлях до файлу ключа у кодуванні UTF-16.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_unmount\_file\_key***

Демонтувати електронний ключ за його ID.

Об'явлення:

```
GohubBool gohub_unmount_file_key(  
    const char* keyId);
```

Параметри:

keyId

Ідентифікатор ключа.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_unmount\_file\_key\_w***

Демонтувати електронний ключ за його ID.

Об'явлення:

```
GohubBool gohub_unmount_file_key_w(  
    const GohubWChar* keyId);
```

Параметри:

keyId

Ідентифікатор ключа у кодуванні UTF-16.

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

## ***gohub\_query\_mounted\_file\_keys***

Запитати кількість монтованих з файлів електронних ключів.

Об'явлення:

```
int gohub_query_mounted_file_keys();
```

Результат:

У разі успіху - не негативна значення. У разі помилки або порожнього списку - 0.  
Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_mounted\_file\_key\_id***

Отримати ID електронного ключа, за його порядковим номером у переліку монтованих електронних ключів.

Об'явлення:

```
const char* gohub_mounted_file_key_id(  
    int index);
```

Параметри:

`index`

Порядковий номер.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_mounted\_file\_key\_id\_w***

Отримати ID електронного ключа (у кодуванні UTF-16), за його порядковим номером у переліку монтованих електронних ключів.

Об'явлення:

```
const GohubWChar* gohub_mounted_file_key_id_w(  
    int index);
```

Параметри:

`index`

Порядковий номер.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_mounted\_file\_key\_dir***

Отримати шлях до файлу електронного ключа за його порядковим номером.

Об'явлення:

```
const char* gohub_mounted_file_key_dir(  
    int index);
```

Параметри:

`index`

Порядковий номер.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_mounted\_file\_key\_dir\_w***

Отримати шлях (у кодуванні UTF-16) до файлу електронного ключа за його порядковим номером.

Об'явлення:

```
const GohubWChar* gohub_mounted_file_key_dir_w(  
    int index);
```

Параметри:

index

Порядковий номер.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### **4.4.14. Робота з АС «Месплан»**

Цей блок функцій дозволяє отримати заявки за вказаним місяцем з АС «Месплан».

<code>gohub_get_mp_months</code>	Запитати перелік ідентифікаторів місяців доступних в АС «Месплан» у вигляді рядку
<code>gohub_get_mp_months_w</code>	Запитати перелік ідентифікаторів місяців доступних в АС «Месплан» у вигляді рядку (в UTF-16)
<code>gohub_query_and_save_orders_for_month</code>	Запитати та зберегти заявки за вказаним місяцем з АС «Месплан» в xml
<code>gohub_query_and_save_orders_for_month_w</code>	Запитати та зберегти заявки за вказаним місяцем з АС «Месплан» в xml (з параметрами в UTF-16)
<code>gohub_query_and_save_orders_for_month_with_relogin</code>	Запитати та зберегти заявки за вказаним місяцем з АС «Месплан» в xml. З можливістю вказати логін та пароль.
<code>gohub_query_and_save_orders_for_month_with_relogin_w</code>	Запитати та зберегти заявки за вказаним місяцем з АС «Месплан» в xml. З можливістю вказати логін та пароль (з параметрами в UTF-16)

## ***gohub\_get\_mp\_months***

Запитати перелік ідентифікаторів місяців доступних в АС «Месплан» у вигляді рядку.

Об'явлення:

```
const char* gohub_get_mp_months(  
    GohubConnection* connection,  
    int codePage)
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

codePage



Числове позначення сторінки кодування, в якій зберігається файл.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_get\_mp\_months\_w***

Запитати перелік ідентифікаторів місяців доступних в АС «Месплан» у вигляді рядку (в UTF-16).

Об'явлення:

```
const GohubWChar* gohub_get_mp_months_w(  
    GohubConnection* connection)
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_query\_and\_save\_orders\_for\_month***

Запитати та зберегти заявки за вказаним місяцем з АС «Месплан» в xml.

Об'явлення:

```
GohubBool gohub_query_and_save_orders_for_month(  
    GohubConnection* connection,  
    const char* month,  
    const char* path)
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`month`

Рядок з номером місяцю за який необхідно запросити заявки.

`Path`

Шлях до файлу в який необхідно зберегти заявки.

Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_query\_and\_save\_orders\_for\_month\_w***

Запитати та зберегти заявки за вказаним місяцем з АС «Месплан» в xml.

Об'явлення:

```
GohubBool gohub_query_and_save_orders_for_month_w(  
    GohubConnection* connection,  
    const char* month,  
    const char* path)
```

```
GohubConnection* connection,  
const GohubWChar* month,  
const GohubWChar* path)
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

month

Рядок з номером місяцю за який необхідно запросити заявки (у кодуванні UTF-16).

Path

Шлях до файлу в який необхідно зберегти заявки (у кодуванні UTF-16).

#### Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_and\_save\_orders\_for\_month\_with\_relogin***

Запитати та зберегти заявки з АС «Месплан» в xml за вказаним місяцем. З можливістю вказати логін та пароль.

#### Об'явлення:

```
GohubBool gohub_query_and_save_orders_for_month(  
GohubConnection* connection,  
const char* month,  
const char* login,  
const char* password,  
const char* path)
```

#### Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

month

Рядок з номером місяцю за який необхідно запросити заявки.

login

Логін до АС «Месплан».

password

Пароль до АС «Месплан».

Path

Шлях до файлу в який необхідно зберегти заявки.

#### Результат:

У разі успіху - значення `True`. У разі помилки - `False`. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## *gohub\_query\_and\_save\_orders\_for\_month\_with\_relogin\_w*

Запитати та зберегти заявки з АС «Месплан» в xml за вказаним місяцем. З можливістю вказати логін та пароль.

Об'явлення:

```
GohubBool gohub_query_and_save_orders_for_month_w(  
    GohubConnection* connection,  
    const GohubWChar* month,  
    const GohubWChar* login,  
    const GohubWChar* password,  
    const GohubWChar* path)
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

month

Рядок з номером місяцю за який необхідно запросити заявки (у кодуванні UTF-16).

login

Логін до АС «Месплан» (у кодуванні UTF-16).

password

Пароль до АС «Месплан» (у кодуванні UTF-16).

Path

шлях до файлу в який необхідно зберегти заявки (у кодуванні UTF-16).

Результат:

У разі успіху - значення True. У разі помилки - False. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### **4.4.15. Робота з документами інформаційних послуг**

Цей блок функцій дозволяє отримати документи інформаційних послуг.

gohub_query_inform_services_document	Запросити документ
gohub_query_next_inform_services_document	Запросити наступний документ
gohub_save_inform_services_document	Зберегти документ
gohub_save_inform_services_document_w	Зберегти документ (з параметрами в UTF-16)
gohub_saveXml_inform_services_document	Зберегти документ в форматі xml
gohub_saveXml_inform_services_document_w	Зберегти документ в форматі xml (з параметрами в UTF-16)
gohub_close_inform_services_document	Закрити документ
gohub_inform_services_document_id	Запросити ID документа
gohub_inform_services_document_revision	Запросити ревізію документа
gohub_inform_services_document_filename	Запитати ім'я документа
gohub_inform_services_document_filename_w	Запитати ім'я документа (в UTF-16)
gohub_inform_services_document_comment	Запитати коментар документа
gohub_inform_services_document_comment_w	Запитати коментар документа (в UTF-16)

gohub_inform_services_document_created_date	Запитати дату створення документа
gohub_inform_services_document_created_date_w	Запитати дату створення документа
gohub_inform_services_document_doc_date	Запитати дату документа
gohub_inform_services_document_doc_date_w	Запитати дату документа (в UTF-16)

## ***gohub\_query\_inform\_services\_document***

Запросити документ інформаційних послуг за ідентифікатором.

Об'явлення:

```
GohubInformServicesDoc* gohub_query_inform_services_document (
    GohubConnection* connection,
    unsigned __int64 docId)
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

docId

Унікальний ідентифікатор документа, що запитується.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_query\_next\_inform\_services\_document***

Запит документа інформаційних послуг наступного за переліком ревізій, починаючи від ревізії що передано параметром.

Об'явлення:

```
GohubInformServicesDoc* gohub_query_next_inform_services_document (
    GohubConnection* connection,
    unsigned __int64 lastRevision)
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції gohub\_connect;

lastRevision

Ревізія документа, від якої починається пошук наступного документа.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

## ***gohub\_save\_inform\_services\_document***

Збереження документа інформаційних послуг (GohubInformServicesDoc) на диск, за вказаним шляхом.

**Об'явлення:**

```
GohubBool gohub_save_inform_services_document(  
    GohubInformServicesDoc* doc,  
    const char* path)
```

**Параметри:**

document

Об'єкт документа (GohubInformServicesDoc), що відправляється на сервер ;

path

Шлях, куди зберігається файл.

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_save\_inform\_services\_document\_w***

Збереження документа інформаційних послуг (GohubInformServicesDoc) на диск, за вказаним шляхом (з параметром в UTF-16).

**Об'явлення:**

```
GohubBool gohub_save_inform_services_document_w(  
    GohubInformServicesDoc* doc,  
    const GohubWChar* path)
```

**Параметри:**

document

Об'єкт документа (GohubInformServicesDoc), що відправляється на сервер ;

path

Шлях, куди зберігається файл (у кодуванні UTF-16).

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_saveXml\_inform\_services\_document***

Збереження документа інформаційних послуг в форматі xml (GohubInformServicesDoc) на диск, за вказаним шляхом.

**Об'явлення:**

```
GohubBool gohub_saveXml_inform_services_document(  
    GohubInformServicesDoc* doc,  
    const char* path)
```

**Параметри:**

document

Об'єкт документа (GohubInformServicesDoc), що відправляється на сервер ;

path

Шлях, куди зберігається файл.

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_saveXml\_inform\_services\_document\_w***

Збереження документа інформаційних послуг в форматі xml (`GohubInformServicesDoc`) на диск, за вказаним шляхом (з параметром в UTF-16).

Об'явлення:

```
GohubBool gohub_saveXml_inform_services_document_w(  
    GohubInformServicesDoc* doc,  
    const GohubWChar* path)
```

Параметри:

`document`

Об'єкт документа (`GohubInformServicesDoc`), що відправляється на сервер ;

`path`

Шлях, куди зберігається файл (у кодуванні UTF-16).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_close\_inform\_services\_document***

Закриття документа інформаційних послуг (`GohubInformServicesDoc`) на диск, за вказаним шляхом (з параметром в UTF-16).

Об'явлення:

```
GohubBool gohub_close_inform_services_document(  
    GohubInformServicesDoc* document)
```

Параметри:

`document`

Об'єкт документа (`GohubInformServicesDoc`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_inform\_services\_document\_id***

Запит ідентифікатору документа інформаційних послуг (`GohubInformServicesDoc`).

Об'явлення:

```
unsigned __int64 gohub_inform_services_document_id(  
    GohubInformServicesDoc* doc)
```

Параметри:

`doc`

Об'єкт документа (`GohubInformServicesDoc`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_inform\_services\_document\_revision***

Запит ревізії документа інформаційних послуг (`GohubInformServicesDoc`).

Об'явлення:

```
unsigned __int64 gohub_inform_services_document_revision(  
    GohubInformServicesDoc* doc)
```

Параметри:

`doc`

Об'єкт документа (`GohubInformServicesDoc`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_inform\_services\_document\_filename***

Запит імені документа інформаційних послуг (`GohubInformServicesDoc`).

Об'явлення:

```
const char* gohub_inform_services_document_filename(  
    GohubInformServicesDoc* document)
```

Параметри:

`document`

Об'єкт документа (`GohubInformServicesDoc`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_inform\_services\_document\_filename\_w***

Запит імені документа інформаційних послуг (`GohubInformServicesDoc`) в UTF-16.

Об'явлення:

```
const GohubWChar* gohub_inform_services_document_filename_w(  
    GohubInformServicesDoc* document)
```

Параметри:

`document`

Об'єкт документа (`GohubInformServicesDoc`) .

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

### ***gohub\_inform\_services\_document\_comment***

Запит коментаря документа інформаційних послуг (`GohubInformServicesDoc`).

Об'явлення:

```
const char* gohub_inform_services_document_comment(  
    GohubInformServicesDoc* document)
```

**Параметри:**

document

Об'єкт документа (GohubInformServicesDoc).

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_inform\_services\_document\_comment\_w***

Запит коментаря документа інформаційних послуг (GohubInformServicesDoc) в UTF-16.

**Об'явлення:**

```
const GohubWChar* gohub_inform_services_document_comment_w(  
    GohubInformServicesDoc* document)
```

**Параметри:**

document

Об'єкт документа (GohubInformServicesDoc).

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_inform\_services\_document\_created\_date***

Запит дати створення документа інформаційних послуг (GohubInformServicesDoc).

**Об'явлення:**

```
const char* gohub_inform_services_document_created_date(  
    GohubInformServicesDoc* document)
```

**Параметри:**

document

Об'єкт документа (GohubInformServicesDoc).

**Результат:**

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін.

### ***gohub\_inform\_services\_document\_created\_date\_w***

Запит дати створення документа інформаційних послуг (GohubInformServicesDoc) в UTF-16.

**Об'явлення:**

```
const GohubWChar* gohub_inform_services_document_created_date_w(  
    GohubInformServicesDoc* document)
```

**Параметри:**

document

Об'єкт документа (GohubInformServicesDoc).



Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_inform\_services\_document\_doc\_date***

Запит дати документа інформаційних послуг (`GohubInformServicesDoc`).

Об'явлення:

```
const char* gohub_inform_services_document_doc_date(  
    GohubInformServicesDoc* document)
```

Параметри:

`document`

Об'єкт документа (`GohubInformServicesDoc`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_inform\_services\_document\_doc\_date\_w***

Запит дати документа інформаційних послуг (`GohubInformServicesDoc`) в UTF-16.

Об'явлення:

```
const GohubWChar* gohub_inform_services_document_doc_date_w(  
    GohubInformServicesDoc* document)
```

Параметри:

`document`

Об'єкт документа (`GohubInformServicesDoc`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## **4.4.16. Обробка помилок**

<code>gohub_last_error_code</code>	Отримати код помилки останньої операції
<code>gohub_last_error_title</code>	Отримати назва помилки останньої операції
<code>gohub_last_error_title_w</code>	Отримати назва помилки останньої операції (в UTF-16)
<code>gohub_last_error_text</code>	Отримати текст помилки останньої операції
<code>gohub_last_error_text_w</code>	Отримати текст помилки останньої операції (в UTF-16)

### ***gohub\_last\_error\_code***

Запит коду помилки виконання останньої операції.

Об'явлення:

```
GohubErrcode gohub_last_error_code();
```

### ***gohub\_last\_error\_title***

Запит заголовка помилки виконання останньої операції.

Об'явлення:

```
const char* gohub_last_error_title();
```

## ***gohub\_last\_error\_title\_w***

Запит заголовка помилки виконання останньої операції (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_last_error_title_w();
```

## ***gohub\_last\_error\_text***

Запит тексту помилки виконання останньої операції.

Об'явлення:

```
const char* gohub_last_error_text();
```

## ***gohub\_last\_error\_text\_w***

Запит тексту помилки виконання останньої операції (у кодуванні UTF-16).

Об'явлення:

```
const GohubWChar* gohub_last_error_text_w();
```

### **4.4.17. Робота з інформацію про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів (за останні 5 днів від поточної дати)**

<code>gohub_query_dispatch_info</code>	Запросити об'єкт з переліком за кодом станцій
<code>gohub_query_dispatch_info_w</code>	Запросити об'єкт з переліком за кодом станцій (в UTF-16)
<code>gohub_document_info_description</code>	Отримати текст опису за індексом
<code>gohub_document_info_description_w</code>	Отримати текст опису за індексом (в UTF-16)
<code>gohub_document_info_count</code>	Отримати кількість значень в переліку
<code>gohub_document_info_is_empty</code>	Отримати текст заповнення вагону
<code>gohub_document_info_is_empty_w</code>	Отримати текст заповнення вагону (в UTF-16)
<code>gohub_document_info_wag_owner</code>	Отримати власника вагону
<code>gohub_document_info_wag_owner_w</code>	Отримати власника вагону (в UTF-16)
<code>gohub_document_info_date</code>	Отримати дату
<code>gohub_document_info_date_w</code>	Отримати дату (в UTF-16)
<code>gohub_document_info_number</code>	Отримати номер
<code>gohub_document_info_number_w</code>	Отримати номер (в UTF-16)
<code>gohub_document_info_type</code>	Отримати тип
<code>gohub_document_info_type_w</code>	Отримати тип (в UTF-16)
<code>gohub_close_dispatch_info</code>	Закриття об'єкту

## ***gohub\_query\_dispatch\_info***

Запросити об'єкт з переліком (GohubDispatchInfo).

Об'явлення:

```
GohubDispatchInfo* gohub_query_dispatch_info(GohubConnection*  
connection, const char* start_esr, const char* end_esr);
```

Параметри:

`connection`

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

`start_esr`

Код ЄМР станції відправлення

end\_esr

Код ЄМР станції призначення

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_query\_dispatch\_info\_w***

Запросити об'єкт з переліком (`GohubDispatchInfo`).

Об'явлення:

```
GohubDispatchInfo* gohub_query_dispatch_info_w(GohubConnection*
connection, const GohubWChar* start_esr, const GohubWChar* end_esr);
```

Параметри:

connection

Показчик з'єднання з Модулем Узгодження, отриманий раніше за допомогою функції `gohub_connect`;

start\_esr

Код ЄМР станції відправлення (в UTF-16)

end\_esr

Код ЄМР станції призначення (в UTF-16)

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_info\_description***

Отримати текст опису за індексом

Об'явлення:

```
const char* gohub_document_info_description(GohubDispatchInfo* document, int
index);
```

Параметри:

document

Об'єкт з переліком (`GohubDispatchInfo`).

index

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_info\_description\_w***

Отримати текст опису за індексом (в UTF-16)

Об'явлення:

```
const GohubWChar* gohub_document_info_description_w(GohubDispatchInfo* document,  
int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_info\_count***

Отримати кількість значень в переліку

Об'явлення:

```
int gohub_document_info_count(GohubDispatchInfo* document);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_info\_is\_empty***

Отримати текст заповнення вагону

Об'явлення:

```
const char* gohub_document_info_is_empty(GohubDispatchInfo* document, int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_info\_is\_empty\_w***

Отримати текст заповнення вагону (в UTF-16)

Об'явлення:

```
const GohubWChar* gohub_document_info_is_empty_w(GohubDispatchInfo* document,  
int index);
```

Параметри:

document

Об'єкт з переліком (GohubDispatchInfo).

index

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_info\_wag\_owner***

Отримати власника вагону

Об'явлення:

```
const char* gohub_document_info_wag_owner(GohubDispatchInfo* document, int index);
```

Параметри:

document

Об'єкт з переліком (GohubDispatchInfo).

index

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_info\_wag\_owner\_w***

Отримати власника вагону(в UTF-16)

Об'явлення:

```
const GohubWChar* gohub_document_info_wag_owner_w(GohubDispatchInfo* document,  
int index);
```

Параметри:

document

Об'єкт з переліком (GohubDispatchInfo).

index

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції gohub\_last\_error і ін .

### ***gohub\_document\_info\_date***

Отримати дату

Об'явлення:

```
const char* gohub_document_info_date(GohubDispatchInfo* document, int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_info\_date\_w***

Отримати дату(в UTF-16)

Об'явлення:

```
const GohubWChar* gohub_document_info_date_w(GohubDispatchInfo* document, int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

### ***gohub\_document\_info\_number***

Отримати номер

Об'явлення:

```
const char* gohub_document_info_number(GohubDispatchInfo* document, int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін.

## ***gohub\_document\_info\_number\_w***

Отримати номер(в UTF-16)

Об'явлення:

```
const GohubWChar* gohub_document_info_number_w(GohubDispatchInfo* document, int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_document\_info\_type***

Отримати тип

Об'явлення:

```
const char* gohub_document_info_type(GohubDispatchInfo* document, int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_document\_info\_type\_w***

Отримати тип(в UTF-16)

Об'явлення:

```
const GohubWChar* gohub_document_info_type_w(GohubDispatchInfo* document, int index);
```

Параметри:

`document`

Об'єкт з переліком (GohubDispatchInfo).

`index`

Індекс за яким отримуємо значення з переліку.

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .

## ***gohub\_close\_dispatch\_info***

Закриття об'єкту

Об'явлення:

```
GohubBool gohub_close_dispatch_info(GohubDispatchInfo* document);
```

Параметри:

`document`

Об'єкт з переліком (`GohubDispatchInfo`).

Результат:

У разі успіху - значення відмінне від нуля. У разі помилки - 0. Інформацію про помилку можна отримати за допомогою функції `gohub_last_error` і ін .



## 4.5. Коди помилок

Коди помилок у *Бібліотеці* представлені переліком (*enum*) `GohubErrcode`, що містить набір констант, семантично відповідних можливим типам помилок, що виникають при роботі з *Бібліотекою*.

*Звідна таблиця кодів помилок*

<code>gohub_success</code>	Операція виконана успішно
<code>gohub_server_connection_could_not_be_established</code>	Помилка при відкритті з'єднання з <i>Модулем Узгодження</i>
<code>gohub_server_inaccessible</code>	Служба <i>Модуля Узгодження</i> недоступна
<code>gohub_document_creation_failed</code>	Помилка відкриття файлу
<code>gohub_document_query_failed</code>	Помилка при запиті документа з АС «Клієнт УЗ» за ідентифікатором
<code>gohub_next_document_query_failed</code>	Помилка при запиті документа з АС «Клієнт УЗ» за номером ревізії
<code>gohub_document_sending_failed</code>	Помилка при відправленні документа в АС «Клієнт УЗ»
<code>gohub_document_saving_failed</code>	Помилка при збереженні документа в файл
<code>gohub_document_loading_failed</code>	Помилка при завантаженні документа з файлу
<code>gohub_invalid_code_page</code>	Недійсна кодова сторінка
<code>gohub_private_key_could_not_be_opened</code>	Помилка при відкритті електронного ключа
<code>gohub_private_key_path_could_not_be_opened</code>	Помилка при відкритті електронного ключа за вказаним шляхом
<code>gohub_private_key_bytes_could_not_be_opened</code>	Помилка при відкритті електронного ключа масивом байт
<code>gohub_private_key_is_inaccessible</code>	Електронний ключ недоступний. Можливо, пристрій від'єднали від комп'ютера
<code>gohub_document_could_not_be_signed</code>	Помилка при накладенні електронно-цифрового підпису
<code>gohub_document_signature_verification_failed</code>	Електронно-цифровий підпис недейсний, можливо, дані були підтверджені
<code>gohub_document_has_not_signature</code>	Документ не містить електронно-цифровий підпис
<code>gohub_client_is_obsolete</code>	Версія клієнта застаріла, необхідно оновити
<code>gohub_server_is_obsolete</code>	Версія сервера застаріла, необхідно оновити
<code>gohub_document_reclamation_failed</code>	Відкликання документа не вдалося
<code>gohub_document_deletion_failed</code>	Видалити документ не вдалося
<code>gohub_attachment_creation_failed</code>	Створити супровідний документ не вдалося
<code>gohub_attachment_sending_failed</code>	Відправити супровідний документ не вдалося
<code>gohub_attachment_query_failed</code>	Запитати супровідний документ не вдалося

gohub_attachment_deletion_failed	Видалити супровідний документ не вдалося
gohub_attaching_to_document_failed	Додати супровідний документ до перевізного документа не вдалося
gohub_detaching_from_document_failed	Відкріпити супровідний документ від перевізного документа не вдалося
gohub_mount_of_file_key_failed	Монтування електронного ключа з файлу не вдалося.
gohub_unmount_of_file_key_failed	Демонтувати електронного ключа з файлу не вдалося.
gohub_enumerating_of_file_keys_failed	Отримати перелік монтованих з файлів електронних ключів не вдалося.
gohub_mounted_file_key_inaccessible	Монтований з файлу електронний ключ не доступний.
gohub_edata_creating_failed	Помилка при створенні ЕД попереднього інформування
gohub_edata_sending_failed	Помилка при відправленні ЕД попереднього інформування в АС «Клієнт УЗ»
gohub_edata Updating_failed	Помилка при оновленні ЕД попереднього інформування в АС «Клієнт УЗ»
gohub_edata_query_failed	Помилка при запиті ЕД попереднього інформування з АС «Клієнт УЗ» за ідентифікатором
gohub_next_edata_query_failed	Помилка при запиті ЕД попереднього інформування з АС «Клієнт УЗ» за ревізією
gohub_pipackage_query_failed	Помилка при запиті пакета ПІ з АС «Клієнт УЗ» за ідентифікатором
Gohub_next_pipackage_query_failed	Помилка при запиті пакета ПІ з АС «Клієнт УЗ» за ревізією
gohub_mp_months_query_failed	Помилка при запиті переліку ідентифікаторів місяців з АС «Месплан»
gohub_orders_of_months_query_failed	Помилка при запиті заявок з АС «Месплан»
gohub_inform_services_doc_saving_failed	Помилка при збереженні документа інформаційних послуг
gohub_inform_services_doc_query_failed	Помилка при запиті документа інформаційних послуг
gohub_query_changes_inform_services_failed	Помилка при запиті змін за документами інформаційних послуг
gohub_query_next_inform_services_document_failed	Помилка при запиті наступного документа інформаційних послуг
gohub_programm_error = 0x1000	Невідома помилка в програмі
gohub_invalid_operation	Програма користувача виконала неприпустиму операцію

Більш детальний опис програмного інтерфейсу бібліотеки можна отримати з файлу gohub.client.errors.h (Додаток Б).

## 4.6. Приклади використання

### а) Вивід на екран інформації про помилку

Визначимо функцію `print_error`, що виводить на екран детальну інформацію про помилку, включно з інформацією про вкладені помилки, якщо вони існують. Ця функція стане в нагоді у наступних прикладах.

```
#include <stdio.h>
#include "gohub.client.h"

void print_last_error()
{
    printf("Error [%04x] -- %s -- %s\n",
        gohub_last_error_code(),
        gohub_last_error_title(),
        gohub_last_error_text());
}
```

### б) Завантаження документа з файлу та передача в АС «Клієнт УЗ»

Зараз продемонструємо завантаження документа з файлу та передачу його в АС «Клієнт УЗ». При цьому використовуємо електронний ключ для накладення електронно-цифрового підпису. Для виводу інформації про можливі помилки використовуємо функцію `print_error`, визначену раніше в п.4.5.а.

```
#include "gohub.client.h"
#include <stdio.h>
#include <string.h>

void print_last_error();

void load_and_send_document(
    const char* host,           // IP адреса або ім'я комп'ютера Модуля Узгодження
    int port,                  // номер TCP порту для підключення
    const char* path,          // шлях до файлу документа
    const char* password) // Пароль до ключу. Якщо не задано, документ не буде
    підписано
{
    GohubConnection* connection;
    GohubDocument* document;
    char document_id[256];

    // Створити підключення до Модулю Узгодження
    printf("Connecting to %s:%d\n", host, port);
    connection = gohub_connect(host, port);

    if(connection)
    { // Підключення створити успішно
        puts("Connection opened");

        // Завантажити документ з файлу
        document = gohub_load_document(path);

        if(document)
        { // Документ завантажено успішно
            printf("Document is loaded: size=%d bytes\n",
                gohub_document_size(document));
            if(password)
            { //Підписати документ
                puts("Signing document . . .");
                if(gohub_open_private_key(connection, password))
```

```

    {
        printf("Key owner name: %s\n",
gohub_private_key_owner_name(connection));
        printf("Private key info: %s\n",
gohub_private_key_owner_info(connection));
        if(gohub_sign_document(connection, document))
            puts("Document signed");
        else
        {
            print_last_error();
            return;
        }
    }
    else
    { //Не вдалося отримати доступ до закритого ключу
        print_last_error();
        return;
    }
}
// Передати документ в АС "Клієнт УЗ" через Модуль Узгодження
if(gohub_send_document(connection, document))
{ // Документ передано успішно
    printf("Document is send: id=%s revision=%d\n",
        gohub_document_id(document),
        gohub_document_revision(document));
    // Перевірка наявності попереджень після відправки документу
    const char* str = gohub_document_warning(doc);
    if (str != NULL)
        printf("Warning in document: %s\n", str);
    // Зберігаємо ідентифікатор документа
    // для подальшого використання
    strcpy(document_id, gohub_document_id(document));
}
else
{ // Помилка при передачі документа
    print_last_error();
}
// Завершити роботу з документом
gohub_close_document(document);
    // Спробуємо запросити цей документ за ідентифікатором
    document = gohub_query_document(connection, document_id);
    if(document)
    { // Запит документа виконано успішно
        // Виводимо документ на екран у кодуванні windows-1251
        puts(gohub_document_text(document));
        // Завершити роботу з документом
        gohub_close_document(document);
    }
    else
    { // Помилка при отриманні документа
        print_last_error();
    }
}
else
{ // Помилка при завантаженні документа з файлу
    print_last_error();
}
// Закрити підключення до Модуля Узгодження
gohub_disconnect(connection);
}
else
{ // Помилка при підключенні до Модулю Узгодження
    print_last_error();
}
}

```

#### в) Запит документів з АС «Клієнт УЗ»

Цей приклад демонструє техніку поступового запиту документів їх АС «Клієнт УЗ» за номерами ревізій. Окрім того показано зміни кодової сторінки документа та Збереження запрошених документів у файли. Окрім того, виконується перевірка електронно-цифрового підпису. Для виводу інформації про помилки по-прежньому використовується функція `print_error`, визначену раніше в п.4.5.а

```
#include "gohub.client.h"
#include <stdio.h>
#include <stdlib.h>

void print_last_error();

int query_and_save_documents(
    const char* host,                // IP адреса або ім'я комп'ютера Модуля
    Узгодження                      // номер TCP порту для підключення
    int port,                        // номер ревізії з якої отримувати перевізні
    int startRevision,              документи
    int maxCount,                  // Максимальна кількість документів , що буде
    const char* targetFolder)       запитано
    // тека куди зберігати отримані документи
{
    int lastRevision = startRevision;
    GohubConnection* connection;
    GohubDocument* document;
    int count;

    // Створити підключення до Модулю Узгодження
    printf("Connectiong to %s:%d", host, port);
    connection = gohub_connect(host, port);

    if(connection)
    { // Підключення створити успішно
        puts("Connection opened");

        for (count = 0; count < maxCount; ++count)
        { // Запит наступного документа за ревізією
            document = gohub_query_next_document(connection, lastRevision);

            if(document)
            { // Запит документа виконано успішно
                printf("Document received: id=%s revision=%d\n",
                    gohub_document_id(document),
                    gohub_document_revision(document));

                // Запам'ятати ревізію останнього документа
                lastRevision = gohub_document_revision(document);

                // Вивести документ на екран
                wprintf(L"%s\n", gohub_document_text_w(document));
                { // Зберегти документ в другой кодировке.
                    char path[_MAX_PATH];
                    sprintf(path, "%s\\%s.xml", targetFolder, gohub_document_id(document));
                    if(gohub_save_document(document, path, 866))
                    { // Документ сохранен успішно
                        printf("Document saved to file: path='%s'\n", path);
                    }
                    else
                    { // Помилка при збереженні документа
                        print_last_error();
                    }
                }
            }
        }
    }
}
```

```

        gohub_close_document(document);
        continue;
    }
}

if(gohub_document_has_signature(document))
{ //Перевірка ЕЦП
    if(gohub_document_check_signature(document))
    {
        wprintf(L"%s\n", gohub_document_signer_name_w(document));
        wprintf(L"%s\n", gohub_document_signer_info_w(document));
        wprintf(L"%s\n", gohub_document_sign_time_w(document));
    }
    else
    {
        print_last_error();
    }
}
// Завершення роботи з документом
gohub_close_document(document);
}
else
{ // Не вдалося отримати наступний за ревізією документ.
  // Це може бути помилка, або нових документів більше немає.
  if(gohub_last_error_code() != gohub_success)
      print_last_error();
  break;
}
}
// Закрити підключення до Модулю Узгодження
gohub_disconnect(connection);
puts("Connection closed");
}
else
{ // Помилка при підключенні до Модулю Узгодження
  print_last_error();
}
// Повернення функцією номера останньої обробленої ревізії
return lastRevision;
}

```

## г) Інші приклади використання

В інсталяційному пакеті Клієнта Модуля Узгодження представлені також інші приклади використання, з якими більш детально можна ознайомитися, заглянув безпосередньо в програмний код (тека Клієнт Модуля Узгодження\samples\ gohub.client.test.c). Тут ми коротко опишемо інші функції, надані у прикладах:

- Продемонстрована техніка поступового запиту документів ФДУ-92 з АС «Клієнт УЗ» за номерами ревізій

```

int query_and_save_fdu92s(
    const char* host,           // IP адреса або ім'я комп'ютера Модуля Узгодження
    int port,                  // номер TCP порту для підключення
    int startRevision,         // номер ревізії з якої отримувати перевірки
    документи
    int maxCount,              // Максимальна кількість документів, що буде
    запитано
    const char* targetFolder); // тека куди зберігати отримані документи

```

- Продемонстрована техніка поступового запиту документів ЕД попереднього інформування з АС «Клієнт УЗ» за номерами ревізій

```

__int64 query_and_save_edatas(
    const char* host,           // IP адреса або ім'я комп'ютера Модуля Узгодження
    int port,                  // номер TCP порту для підключення
    __int64 startRevision,     // номер ревізії з якої отримувати ЕД (електронні
дані) попереднього інформування
    int maxCount,              // Максимальна кількість документів ЕД, що буде
запитано
    const char* targetFolder); // тека куди зберігати отримані документи ЕД

```

- Продемонстрована техніка поступового запиту пакетів попереднього інформування з АС «Клієнт УЗ» за номерами ревізій

```

__int64 query_and_save_pi_packages(
    const char* host,           // IP адреса або ім'я комп'ютера Модуля Узгодження
    int port,                  // номер TCP порту для підключення
    __int64 startRevision,     // номер ревізії з якої отримувати пакети ПІ
(попереднього інформування)
    int maxCount,              // Максимальна кількість пакети ПІ, що буде
запитано
    const char* targetFolder); // тека куди зберігати отримані пакети ПІ

```

- Продемонстрована техніка додавання документа ЕД в пакет попереднього інформування

```

void add_edata_to_pi_package(
    const char* host,           // IP адреса або ім'я комп'ютера Модуля Узгодження
    int port,                  // номер TCP порту для підключення
    const char* piPackageId,   // ID пакета ПІ (попереднього інформування), в який
будуть додаватися ЕД (електронні дані)
    const char* edataPath,     // шлях до xml-файлу з ЕД
    int edataType,             // код типу ЕД: 190 рахунок-фактура, 320
пакувальний лист
    char *newEdataId);         // сюди буде отримано ID створеного документа ЕД

```

- Продемонстрована техніка оновлення документа ЕД, що міститься на сервері АС «Клієнт УЗ»

```

__int64 update_edata(
    const char* host,           // IP адреса або ім'я комп'ютера Модуля Узгодження
    int port,                  // номер TCP порту для підключення
    const char* edataId,       // ID документа ЕД що буде змінюватись
    const char* edataPath);    // шлях до xml-файлу з ЕД

```

- Продемонстрована техніка отримання переліку з інформацією про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів(за останні 5 днів від поточної дати)

```

try
{
    printf(" Test Query Dispatch Info(C++)\n");
    printf(" Please enter ESR number of start station: ");
    char start_esr[101];
    scanf("%s", start_esr);
    printf(" Please enter ESR number of end station: ");
    char end_esr[101];
    scanf("%s", end_esr);
    GohubDispatchInfo* str = gohub_query_dispatch_info(connection,
start_esr, end_esr);
    check_errors();
    printf("\nlist loaded.\n");
    for(int i = 0; i < gohub_document_info_count(str); i++)
    {
        printf("%s\n", gohub_document_info_description(str, i));
        printf("%s\n", gohub_document_info_number(str, i));
    }
}

```

```
        printf("%s\n", gohub_document_info_date(str, i));
        printf("%s\n", gohub_document_info_type(str, i));
        printf("%s\n", gohub_document_info_wag_owner(str, i));
        printf("%s\n", gohub_document_info_is_empty(str, i));
    }
    gohub_close_dispatch_info(str);
    printf("-----\n");
}
catch (...)
{
    printf("Exception");
}
```



## 5. .NET бібліотека - TMSoft.Gohub.Client.Net.dll

Програмний інтерфейс *Бібліотеки* надано набором типів, перелік яких показано нижче. Усі типи бібліотеки визначені у просторі імен *TMSoft.Gohub.Client*.

### 5.1. Перелік типів

GohubConnection	Клас з'єднань з Сервером Модуля Узгодження
GohubDocument	Клас перевізних документів
GohubAttachment	Клас супровідних документів
GohubEData	Клас супровідних документів
GohubInformServicesDoc	Клас документів інформаційних послуг
GohubSigner	Клас інформації про власників електронних ключів та авторів електронно-цифрових підписів
GohubDocumentFilter	Клас інформації про стани фільтру запиту документів
GohubException	Клас виключень
GohubErrCode	Перелік кодів помилок
DispatchInfo	Клас № Замовлення для перевізного документу

### 5.2. Стани перевізного документа

Коди перевізного документа в *Бібліотеці* представлені переліком (*enum*) *DocumentStatus*, що містить набір констант.

*Звідна таблиця кодів станів перевізного документа*

Unknown = 0	Статус невідомий
Draft = 1	Чернетка
Sending = 2	Документ передається товарному касиру
Registered = 3	Документ переданий товарному касиру
Reclaiming = 4	Документ відкликається від товарного касира
Accepted = 5	Вантаж прийнято до перевезення
Delivered = 6	Вантаж прибув
Recieved = 7	Вантаж отримано одержувачем
Uncredited = 8	Документ розкредитовано товарним касиром
RecDraft = 9	Вантаж отримано одержувачем і редагується
RecSending = 10	Вантаж отримано одержувачем і переданий товарному касиру
RecReclaiming = 11	Вантаж отримано одержувачем і відкликається від товарного касира
Canceled = 12	Документ зіпсований товарним касиром

### 5.3. GohubConnection

Клас *GohubConnection* використовується для представлення з'єднань з Сервером Модуля Узгодження та містить наступні члени:

GohubConnection (конструктор)	Створити з'єднання з Сервером Модуля Узгодження. Для закриття з'єднання необхідно використати метод <i>Dispose</i> .
----------------------------------	--

DocumentFilter (властивість)	Екземпляр класу GohubDocumentFilter, що дозволяє встановлювати фільтри для запиту документів.
SignerInfo (властивість)	Інформація про власника електронного, відкритого за допомогою метода OpenPrivateKey. У випадку, якщо пристрій електронного ключа було відключено, генерується виключення GohubException з кодом помилки GohubErrCode.gohub_private_key_is_inaccessible. У випадку, якщо електронний ключ не було відкрито за допомогою метода OpenPrivateKey, генерується виключення GohubException з кодом помилки GohubErrCode.gohub_invalid_operation
ClosePrivateKey (метод)	Закрити електронний ключ, відкритий раніше за допомогою метода OpenPrivateKey. Електронний ключ автоматично закривається також при закритті з'єднання з Сервером Модуля Узгодження.
Dispose (метод)	Закрити з'єднання з Сервером Модуля Узгодження. Автоматично також закривається електронний ключ, якщо його було відкрито.
OpenPrivateKey (метод)	Відкрити електронний ключ. Якщо електронний ключ було відкрито уже раніше, він автоматично закривається. Електронний ключ автоматично закривається також при закритті з'єднання з Сервером Модуля Узгодження. Для закриття електронного ключа без розриву з'єднання з Сервером Модуля Узгодження можна використати метод ClosePrivateKey. Електронний ключ потрібен для підписання електронних документів. Для перевірки електронно-цифрового підпису електронний ключ не використовується та його можна не відкривати. Можливі наступні варіанти відкриття електронного ключа: <ul style="list-style-type: none"> <li>• За паролем. Ключ повинен бути монтований.</li> <li>• За вказаним шляхом та паролю.</li> <li>• Ключ надано масивом байт. Необхідно вказати пароль та довжину масиву байт.</li> </ul>
QueryDocument (метод)	Запросити документ за унікальним ідентифікатором.
QueryDocuments (метод)	Отримати впорядковану послідовність документів, з номерами ревізії більше заданої
QueryDocumentPrintableForm (метод)	Запитати друковану форму документа за його ID. Повертає масив байт.
QueryAndSaveDocumentPrintableForm (метод)	Запитати друковану форму документа за його ID. Результат запиту зберігається в файл за шляхом, що вказано в параметрі path.
SendDocument (метод)	Надіслати документ в АС «Клієнт УЗ»
SignDocument (метод)	Підписати документ електронно-цифровим підписом
ReclaimDocument (метод)	Відкликання документа з сервера СГР
DeleteDocument (метод)	Видалення документа на АС «Клієнт УЗ» за його ID
SendAttachment (метод)	Відправити супровідний документ в АС «Клієнт УЗ»
QueryAttachment (метод)	Запитати супровідний документ з системи АС «Клієнт УЗ»
QueryAttachmentWithUserData (метод)	Запитати супровідний документ з електронними даними користувача з системи АС «Клієнт УЗ»
DeleteAttachment (метод)	Видалити супровідний документ з АС «Клієнт УЗ»
QueryEData	Запитати електронні дані ПІ з системи АС «Клієнт УЗ»

(метод)	
QueryEDatas (метод)	Запитати електронні дані ПІ з системи АС «Клієнт УЗ» за ревізією
SendEData (метод)	Відправити електронні дані ПІ в АС «Клієнт УЗ»
AddEDataToPiPackage (метод)	Додати електронні дані ПІ в АС «Клієнт УЗ»
UpdateEData (метод)	Оновити наявні електронні дані ПІ в АС «Клієнт УЗ»
QueryPiPackage (метод)	Запитати пакет ПІ з системи АС «Клієнт УЗ»
QueryPiPackages (метод)	Запитати пакети ПІ з системи АС «Клієнт УЗ» за ревізією
SendReceivedDocument (метод)	Надіслати документ по прибуттю
QueryMPMonths (метод)	Запитати перелік ідентифікаторів місяців з АС «Месплан» у вигляді переліку
QueryMPMonthsString (метод)	Запитати перелік ідентифікаторів місяців з АС «Месплан» у вигляді рядку (номера місяців відокремлені пробілом)
QueryOrdersForMonth (метод)	Запитати заявки за номером місяцю з АС «Месплан». Можливий запит з вказанням логону та пароля.
QueryAndSaveOrdersForMonth (метод)	Запитати та зберегти заявки за номером місяцю в xml. Можливий запит з вказанням логону та пароля.
QueryEDataForAttachment (метод)	Запитати електронні дані ПІ з системи АС «Клієнт УЗ» за ідентифікатором супровідного документа
QueryAndSaveFdu92PrintableForm (метод)	Запитати друковану форму ФДУ-92 за його ID. Результат запиту зберігається в файл за шляхом, що вказано в параметрі path.
QueryFdu92PrintableForm (метод)	Запитати друковану форму ФДУ-92 за його ID. Повертає масив байт.
QueryAndSaveGu46PrintableForm (метод)	Запитати друковану форму ГУ-46 за його ID. Результат запиту зберігається в файл за шляхом, що вказано в параметрі path.
QueryGu46PrintableForm (метод)	Запитати друковану форму ГУ-46 за його ID. Повертає масив байт.
QueryAndSaveGu45PrintableForm (метод)	Запитати друковану форму ГУ-45 за його ID. Результат запиту зберігається в файл за шляхом, що вказано в параметрі path.
QueryGu45PrintableForm (метод)	Запитати друковану форму ГУ-45 за його ID. Повертає масив байт.
QueryFdu92 (метод)	Запитати ФДУ-92 за унікальним ідентифікатором.
QueryFdu92_ByNumber (метод)	Запитати ФДУ-92 за номером станції та номером накопичувальної картки
QueryFdu92s (метод)	Отримати впорядковану послідовність ФДУ-92, з номерами ревізії більше заданої
SignFdu92 (метод)	Підписати накопичувальну картку електронно-цифровим підписом ФДУ-92
SendFdu92 (метод)	Відправити ФДУ-92 в АС «Клієнт УЗ»
QueryGu46 (метод)	Запитати ГУ-46 за унікальним ідентифікатором.
QueryGu46s (метод)	Отримати впорядковану послідовність ГУ-46, з номерами ревізії більше заданої
SignGu46 (метод)	Підписати накопичувальну картку електронно-цифровим підписом ГУ-46
SendGu46 (метод)	Відправити ГУ-46 АС «Клієнт УЗ»
QueryGu45 (метод)	Запитати ГУ-45 за унікальним ідентифікатором.
QueryGu45s	Отримати впорядковану послідовність ГУ-45, з номерами ревізії

(метод)	більше заданої
QueryInformServicesDoc (метод)	Запросити документ інформаційних послуг за унікальним ідентифікатором.
QueryChangesInformServices (метод)	Отримати впорядковану послідовність документів інформаційних послуг, з номерами ревізії більше заданої
QueryDispatchInfo (метод)	Отримати переліз з інформацією про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів(за останні 5 днів від поточної дати)

## 5.4. GohubDocument

Клас GohubDocument використовується для представлення електронних перевізних документів та містить наступні члени:

HasSignature (властивість)	Ознака наявності у документа електронно-цифрового підпису
Id (властивість)	Унікальний ідентифікатор документа
Revision (властивість)	Номер ревізії документа
Status (властивість)	Статус документа
SignerInfo (властивість)	Інформація про особу, що підписала документ електронно-цифровим підписом. Якщо перевірка електронно-цифрового підпису не підтверджує достовірність даних, результатом буде null. Якщо документ не містить електронно-цифрового підпису, генерується виключення GohubException з кодом помилки GohubErrCode.gohub_document_has_not_signature.
TimeStamp (властивість)	Позначка часу, отримана при підписанні документа електронно-цифровим підписом. Якщо перевірка електронно-цифрового підпису не підтверджує достовірність даних, генерується виключення GohubException з кодом помилки GohubErrCode.gohub_document_signature_verification_failed. Якщо документ не містить електронно-цифрового підпису, генерується виключення GohubException з кодом помилки GohubErrCode.gohub_document_has_not_signature.
Attachments (властивість)	Перелік ID супровідних документів, доданих до цього документа.
MeasureEquipNum (властивість get; set;)	Відомості вагонівимірювальної техніки.
BusinessUnitNum (властивість get; set;)	Номер філіалу ПрАТ УЗ.
SetVerifiedEmptyWeightForWagon (метод)	Встановити уточнену вагу тари вагона <i>int wagonIndex</i> – індекс вагона уточнену вагу якого необхідно встановити. Індикація вагонів починається з нуля. <i>int weight</i> – уточнену вагу вагона.
GetVerifiedEmptyWeightForWagon (метод)	Отримати уточнену вагу тари вагона <i>int wagonIndex</i> – індекс вагона уточнену вагу якого необхідно отримати. Індикація вагонів починається з нуля.
ForeignNotAccept (властивість get;)	Отримати позначку повернення неприйнятих прикордонними станціями іноземних залізниць вагонів на територію України
WarrantType (властивість get; set;)	Тип підстав для отримання вантажу (0 - довіреність, 1 - наказ)
Deserialize (методи)	Десеріалізувати документ з масиву байт, потоку вводу, TextReader або XmlReader. У випадку десеріалізації з масиву байт або потоку вводу Xml-

	<p>документ, повинен починатися з xml-заголовка з вказанням кодування xml-документа. Фактичне кодування символів xml-документа повинне відповідати задекларований в заголовку. У випадку відсутності заголовка xml-документа, по замовченню вважається, що xml-документ надано у кодуванні utf-8.</p> <p>У випадку десеріалізації документа за допомогою TextReader або XmlReader заголовок не є обов'язковим та ігнорується як надлишкова інформація.</p>
FromXml (методы)	Створити документ з XML-документа (XmlDocument) або його елементу (XmlElement)
FromXmlText (метод)	Створити документ з тексту XML-документа
GetXmlText (метод)	Отримати XML-текст документа
GetDataXmlText (метод)	<p>Отримати XML-текст електронних даних документа в заданій версії ЕПД</p> <p><i>int epdVersion</i> - версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4)</p>
Load (метод)	Завантажити документ з файлу. Xml-документ, що міститься у файлі повинен починатися з xml-заголовка з вказанням кодування xml-документа. Фактичне кодування символів xml-документа повинна відповідати задекларований в заголовку. У випадку відсутності заголовка xml-документа, по замовченню вважається, що xml-документ надано у кодуванні utf-8.
Save (метод)	Зберегти документ в файл у заданому кодуванні
SaveData (метод)	<p>Зберегти електронні дані документа в файл у заданому кодуванні та в заданій версії ЕПД</p> <p><i>int epdVersion</i> - версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4)</p>
Serialize (методы)	Серіалізувати документ в масив байт, потоку виводу, TextWriter або XmlWriter
SerializeData (методы)	<p>Серіалізувати електронні дані документа в масив байт, потоку виводу, TextWriter або XmlWriter в заданій версії ЕПД</p> <p><i>int epdVersion</i> - версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4)</p>
ToXml (метод)	Перетворити електронний перевізний документ в XML-документ (XmlDocument)
DataToXml (метод)	<p>Перетворити електронні дані документа в XML-документ (XmlDocument) в заданій версії ЕПД</p> <p><i>int epdVersion</i> - версія ЕПД з якою можна отримати електронні дані документа (прийняті значення 10, 11, 12, 13, 14, що відповідає версії ЕПД 1.0, 1.1, 1.2, 1.3, 1.4)</p>
GetOTPR (метод)	Збереження актуального тексту перевізного документа у форматі xml в файл.
GetOTPRString (метод)	Отримати актуальний текст перевізного документа у кодуванні utf-8.
Warning (властивість)	Текст попередження після відправки документу

## 5.5. GohubAttachment

Клас `GohubAttachment` використовується для представлення інформації про супровідних документах та містить наступні члени:

<code>Id</code> (властивість)	Унікальний ідентифікатор супровідного документа
<code>Description</code> (властивість)	Текстовий опис супровідного документа, що формується з назви типу документа, його реєстраційного номера та дати.
<code>Name</code> (властивість)	Назва супровідного документа.
<code>TypeCode</code> (властивість)	Код типу супровідного документа для бланків ГУ та ЦІМ.
<code>SmgsTypeCode</code> (властивість)	Код типу супровідного документа для бланків СМГС та ЦІМ/СМГС згідно інформаційного керівництва СМГС.
<code>RegistrationNumber</code> (властивість)	Реєстраційний номер супровідного документа.
<code>RegistrationDate</code> (властивість)	Дата реєстрації супровідного документа.
<code>ValidFrom</code> (властивість)	Дата початку дії супровідного документа.
<code>ValidTo</code> (властивість)	Дата припинення дії супровідного документа.
<code>Load</code> (метод)	Створює проєкт <code>GohubAttachment</code> по параметрам: <i>string typeCode</i> – код типу документа для бланків ГУ та ЦІМ; <i>string smgsTypeCode</i> – код типу документа для бланків СМГС та ЦІМ/СМГС; <i>string name</i> – назва документа; <i>string registrationNumber</i> – реєстраційний номер; <i>System.DateTime registrationDate</i> – дата реєстрації; <i>System.DateTime validFrom</i> – дата, з якої супровідний документ дійсний; <i>System.DateTime validTo</i> – дата, до якої супровідний документ дійсний; <i>string path</i> – шлях до відсканованого тексту документа в pdf форматі.
<code>Load</code> (метод)	Створює проєкт <code>GohubAttachment</code> по параметрам: <i>string typeCode</i> – код типу документа для бланків ГУ та ЦІМ; <i>string smgsTypeCode</i> – код типу документа для бланків СМГС та ЦІМ/СМГС; <i>string name</i> – назва документа; <i>string registrationNumber</i> – реєстраційний номер; <i>System.DateTime registrationDate</i> – дата реєстрації; <i>System.DateTime validFrom</i> – дата, з якої супровідний документ дійсний; <i>System.DateTime validTo</i> – дата, до якої супровідний документ дійсний; <i>string path</i> – шлях до відсканованого тексту документа в pdf форматі. <i>string pathUserData</i> – шлях до електронних даних супровідного документа в xml форматі.
<code>Save</code> (метод)	Зберігає відсканований текст супровідного документа в pdf форматі. <i>string path</i> – шлях до відсканованого тексту документа в pdf форматі.
<code>SaveUserData</code> (метод)	Зберігає дані користувача супровідного документа в xml форматі. <i>string path</i> – шлях до електронних даних супровідного документа в xml форматі.

## 5.6. GohubEData

Клас `GohubEData` використовується для представлення інформації про електронних даних ПІ та містить наступні члени:

<code>Id</code> (властивість)	Унікальний ідентифікатор електронних даних ПІ
<code>Revision</code> (властивість)	Номер ревізії електронних даних ПІ.
<code>RevisionDate</code> (властивість)	Дата ревізії електронних даних ПІ.
<code>Version</code> (властивість)	Версія електронних даних ПІ.
<code>DocType</code> (властивість)	Код типу електронних даних ПІ: 190 рахунок-фактура, 320 пакувальний лист.
<code>Status</code> (властивість)	Код статусу електронних даних ПІ.
<code>AttachmentId</code> (властивість)	ID супровідного документа електронних даних ПІ.
<code>Data</code> (властивість)	Масив даних що міститься в електронних даних ПІ.
<code>Load</code> (метод)	Створює проєкт <code>GohubEData</code> по параметрам: <i>ulong attachmentSmgsTypeCode</i> – код супровідного документа: 325 Рахунок-проформа (Інвойс), 380 (Інвойс) рахунок-фактура, 935 Рахунок-фактура; <i>string xmlPath</i> – шлях до xml-файлу що містять електронні дані ПІ; <i>string name</i> – назва супровідного документа; <i>string registrationNumber</i> – реєстраційний номер; <i>System.DateTime registrationDate</i> – дата реєстрації; <i>System.DateTime validFrom</i> – дата, з якої супровідний документ дійсний; <i>System.DateTime validTo</i> – дата, до якої супровідний документ дійсний; <i>string pdfPath</i> – шлях до відсканованого тексту супровідного документа в pdf форматі.
<code>Load</code> (метод)	Створює проєкт <code>GohubEData</code> по параметрам: <i>ulong attachmentSmgsTypeCode</i> – код супровідного документа: 325 Рахунок-проформа (Інвойс), 380 (Інвойс) рахунок-фактура, 935 Рахунок-фактура; <i>string xmlPath</i> – шлях до xml-файлу що містять електронні дані
<code>Save</code> (метод)	Зберігає текст електронних даних ПІ в xml форматі. <i>string path</i> – шлях до тексту електронних даних ПІ в xml форматі.
<code>LoadData</code> (метод)	Завантажити дані з файлу у наявні електронні дані ПІ. <i>string path</i> – шлях до тексту електронних даних ПІ в xml форматі.

## 5.7. GohubPiPackage

Клас `GohubPiPackage` використовується для представлення інформації пакету ПІ та містить наступні члени:

<code>Id</code> (властивість)	Унікальний ідентифікатор пакета ПІ
<code>Revision</code> (властивість)	Номер ревізії пакета ПІ.
<code>RevisionDate</code> (властивість)	Дата ревізії пакета ПІ.

ConsignmentId (властивість)	ID перевізного документа на який посилається пакет ПІ.
Status (властивість)	Код статусу пакета ПІ.
PiPackageToEDataList (властивість)	Перелік GohubPiPackageToEData у пакеті ПІ.
Data (властивість)	Масив даних що міститься в повній інформації про пакету ПІ.
Save (метод)	Зберігає текст дані пакета ПІ в xml форматі. <i>string path</i> – шлях до тексту даних пакета ПІ в xml форматі.

## 5.8. GohubPiPackageToEData

Клас GohubPiPackageToEData використовується для представлення інформації про електронних даних що містяться у пакеті ПІ та містить наступні члени:

Id (властивість)	ID про'єкту GohubPiPackageToEData
EDataId (властивість)	ID електронних даних в про'єкті GohubPiPackageToEData
EDataVersion (властивість)	Версія електронних даних в про'єкті GohubPiPackageToEData
PiPackageId (властивість)	ID пакета ПІ в про'єкті GohubPiPackageToEData
Status (властивість)	Статус про'єкту GohubPiPackageToEData
Note (властивість)	Примітки про'єкту GohubPiPackageToEData

## 5.9. GohubFdu92

Клас GohubFdu92 використовується для представлення інформації про накопичувальні картки ФДУ-92 та містить наступні члени:

Id (властивість)	Унікальний ідентифікатор накопичувальної картки ФДУ-92
Revision (властивість)	Номер ревізії накопичувальної картки ФДУ-92.
Status (властивість)	Статус накопичувальної картки ФДУ-92.
HasSignature (властивість)	Властивість, що перевіряє чи підписано накопичувальну картку ФДУ-92 електронно-цифровим підписом.
SignerInfo (властивість)	Інформація про особу, що підписала накопичувальну картку ФДУ-92 електронно-цифровим підписом.
TimeStamp (властивість)	Позначка часу, отримана при підписанні накопичувальної картки ФДУ-92 електронно-цифровим підписом.
Load (метод)	Завантажити накопичувальну картку ФДУ-92 з файлу Xml.
Save (метод)	Зберігає текст накопичувальної картки ФДУ-92 в Xml форматі.
GetXmlText (властивість)	Отримати ФДУ-92 у вигляді рядку
ToXml (властивість)	Перетворити ФДУ-92 в Xml-документ



## 5.10. GohubGu46

Клас `GohubGu46` використовується для представлення інформації про відомості користування вагонами/контейнерами ГУ-46 та містить наступні члени:

<code>Id</code> (властивість)	Унікальний ідентифікатор відомості ГУ-46
<code>Revision</code> (властивість)	Номер ревізії відомості ГУ-46
<code>Status</code> (властивість)	Статус відомості ГУ-46
<code>HasSignature</code> (властивість)	Властивість, що перевіряє, чи підписано відомість ГУ-46 електронно-цифровим підписом.
<code>SignerInfo</code> (властивість)	Інформація про особу, що підписала відомість ГУ-46 електронно-цифровим підписом.
<code>TimeStamp</code> (властивість)	Позначка часу, отримана при підписанні відомості ГУ-46 Електронно-цифровим підписом.
<code>Load</code> (метод)	Завантажити відомість ГУ-46 з файлу Xml
<code>Save</code> (метод)	Зберігає текст відомості ГУ-46 в Xml форматі
<code>GetXmlText</code> (властивість)	Отримати ГУ-46 у вигляді рядку
<code>ToXml</code> (властивість)	Перетворити ГУ-46 в Xml-документ

## 5.11. GohubGu45

Клас `GohubGu45` використовується для представлення інформації про пам'ятку подання/прибирання вагонів та видачу/прийом контейнерів ГУ-45 та містить наступні члени:

<code>Id</code> (властивість)	Унікальний ідентифікатор пам'ятки ГУ-45
<code>Revision</code> (властивість)	Номер ревізії пам'ятки ГУ-45
<code>Status</code> (властивість)	Статус пам'ятки ГУ-45
<code>HasSignature</code> (властивість)	Властивість, що перевіряє, чи підписано пам'ятку ГУ-45 електронно-цифровим підписом.
<code>SignerInfo</code> (властивість)	Інформація про особу, що підписала пам'ятку ГУ-45 електронно-цифровим підписом.
<code>TimeStamp</code> (властивість)	Позначка часу, отримана при підписанні пам'ятки ГУ-45 електронно-цифровим підписом.
<code>Save</code> (метод)	Зберігає текст пам'ятки ГУ-45 в Xml форматі
<code>GetXmlText</code> (властивість)	Пам'ятка про подачу / прибирання вагонів у вигляді рядку
<code>ToXml</code> (властивість)	Перетворити ГУ-45 в Xml-документ

## 5.12. GohubDocumentFilter

Клас `GohubDocumentFilter` використовується для керування фільтрами для запитів документів з сервера та містить наступні члени:

<code>WagonNumber</code>	Фільтрація за номером вагона
--------------------------	------------------------------

(властивість)	
DocumentNumber (властивість)	Фільтрація за номером документа
DocumentStatus (властивість)	Фільтрація за статусом документа
DepartureClientCode (властивість)	Фільтрація за кодом відправника
DeparturePayerCode (властивість)	Фільтрація за кодом платника по відправленню
DepartureStationCode (властивість)	Фільтрація за кодом станції відправлення
ArrivalClientCode (властивість)	Фільтрація за кодом одержувача
ArrivalPayerCode (властивість)	Фільтрація за кодом платника по прибуттю
ArrivalStationCode (властивість)	Фільтрація за кодом станції призначення
Clear (метод)	Очистить всі фільтри.

## 5.13. GohubSigner

Клас `GohubSigner` використовується для представлення інформації про власників електронних ключів, авторів електронно-цифрових підписів та містить наступні члени:

Address (властивість)	Адрес
Department (властивість)	Відділ
Dns (властивість)	DNS
DrfoCode (властивість)	Код ДРФО
Edrpou_code (властивість)	Код ЕДРПОУ
Email (властивість)	Електронний адреса
Establishment (властивість)	Установа
FullName (властивість)	Повне ім'я (прізвище, ім'я, по батькові)
Issuer (властивість)	Організація, що видала сертифікат справжності
IssuerSummary (властивість)	Повна інформація про організацію, що видала сертифікат справжності
Locality (властивість)	Місто
Name (властивість)	Ім'я
Phone (властивість)	Телефон
Region (властивість)	Область
SerialNumber (властивість)	Серійний номер сертифікату
StaffPost (властивість)	Поштовий індекс
Summary (властивість)	Підсумкова інформація
ToString (властивість)	Отримати рядкове представлення інформації про власника електронного ключа

## 5.14. GohubClient

Клас `GohubClient` використовується для підключення файлів електронних ключів та містить наступні члени:

<code>MountFileKey</code> (метод)	Монтувати файл електронного ключа, указав шлях до директорії та указав ідентифікатор для ключа
<code>UnmountFileKey</code> (метод)	Демонтувати файл електронного ключа, за його ідентифікатором
<code>GetMountedKeys</code> (метод)	Отримати масив ідентифікаторів монтованих файлів електронних ключів
<code>GetMountedKeyDir</code> (метод)	Отримати шлях до директорії в якій знаходиться файл електронного ключа за його ідентифікатором

## 5.15. GohubException

Клас `GohubException` використовується для інформування про помилки часу виконання з використанням механізму виключень платформи .NET Framework та містить наступні члени:

<code>ErrCode</code> (властивість)	Код помилки. Можливі значення див. в п.5.6 <code>GohubErrCode</code>
---------------------------------------	--

## 5.16. GohubErrCode

Перелік `GohubErrCode` уявляє собою набір кодів помилок та містить наступні члени:

<code>gohub_success = 0</code>	Операція виконана успішно
<code>gohub_server_connection_could_not_be_established = 1</code>	Неможливо встановити з'єднання з Сервером Модуля Узгодження
<code>gohub_server_inaccessible = 2</code>	Сервер Модуля Узгодження недоступний
<code>gohub_document_creation_failed = 3</code>	Неможливо створити документ
<code>gohub_document_query_failed = 4</code>	Запитати документ за ідентифікатором не вдалося
<code>gohub_next_document_query_failed = 5</code>	Запитати документ за ревізією не вдалося
<code>gohub_document_sending_failed = 6</code>	Відправити документ в АС «Клієнт УЗ» не вдалося
<code>gohub_document_saving_failed = 7</code>	Зберегти документ не вдалося
<code>gohub_document_loading_failed = 8</code>	Завантажити документ не вдалося
<code>gohub_private_key_could_not_be_opened = 10</code>	Не вдалося відкрити електронний ключ
<code>gohub_private_key_is_inaccessible = 11</code>	Електронний ключ недоступний
<code>gohub_document_could_not_be_signed = 12</code>	Не вдалося підписати документ
<code>gohub_document_signature_verification_failed = 13</code>	Перевірка електронно-цифрового підпису не підтвердила достовірність даних
<code>gohub_document_has_not_signature = 14</code>	Неприпустима операція. Документ не містить електронно-цифрового підпису
<code>gohub_client_is_obsolete = 15</code>	Версія клієнта застаріла, необхідно оновити
<code>gohub_server_is_obsolete = 16</code>	Версія сервера застаріла, необхідно оновити
<code>gohub_document_reclamation_failed = 17</code>	Відкликати документ не вдалося
<code>gohub_document_deletion_failed = 18</code>	Видалити документ не вдалося

gohub_attachment_creation_failed = 19	Створити супровідний документ не вдалося
gohub_attachment_sending_failed = 20	Відправити супровідний документ не вдалося
gohub_attachment_query_failed = 21	Запитати супровідний документ не вдалося
gohub_attachment_deletion_failed = 22	Видалити супровідний документ не вдалося
gohub_attaching_to_document_failed = 23	Додати супровідний документ до перевізного документу не вдалося
gohub_detaching_from_document_failed = 24	Відкріпити супровідний документ від перевізного документа не вдалося
gohub_mount_of_file_key_failed = 25	Монтування електронного ключа з файлу не вдалося.
gohub_unmount_of_file_key_failed = 26	Демонтувати електронного ключа з файлу не вдалося.
gohub_enumerating_of_file_keys_failed = 27	Отримати перелік монтованих з файлів електронних ключів не вдалося.
gohub_mounted_file_key_inaccessible = 28	Монтований з файлу електронний ключ не доступний.
gohub_edata_creation_failed = 29	Неможливо створити ЕД ПІ
gohub_edata_sending_failed = 30	Оправити ЕД ПІ не вдалося
gohub_edata Updating_failed = 31	Оновити ЕД ПІ не вдалося
gohub_edata_query_failed = 32	Запит ЕД ПІ за ідентифікатором не вдалося
gohub_next_edata_query_failed = 33	Запит ЕД ПІ за ревізією не вдалося
gohub_add_edata_to_pipackage_failed = 34	Додати ЕД в пакет ПІ не вдалося
gohub_pipackage_query_failed = 35	Запит пакета ПІ за ідентифікатором не вдалося
gohub_next_pipackage_query_failed = 36	Запит пакета ПІ за ревізією не вдалося
gohub_mp_months_query_failed = 37	Помилка при запиті переліку ідентифікаторів місяців з АС «Месплан»
gohub_orders_of_months_query_failed = 38	Помилка при запиті заявок з АС «Месплан»
gohub_programm_error = 4096	Невідома помилка в програмі
gohub_invalid_operation = 4097	Клієнтський застосунок виконав неприпустиму операцію
gohub_inform_services_doc_saving_failed = 4098	Збереження документа інформаційних послуг не вдалося
gohub_inform_services_doc_query_failed = 4099	Запит документа інформаційних послуг за ідентифікатором не вдалося
gohub_query_changes_inform_services_failed = 4100	Запит ідентифікаторів документів інформаційних послуг з ревізіями вище зазначеної не вдалося
gohub_query_next_inform_services_document_failed = 4101	Запит наступного згідно ревізії документа інформаційних послуг не вдалося
gohub_gu45_get_xml_text_failed	Отримати ГУ-45 у вигляді рядку не вдалося
gohub_gu45_to_xml_failed	Отримати ГУ-45 у вигляді XmlDocument не вдалося
gohub_gu46_get_xml_text_failed	Отримати ГУ-46 у вигляді рядку не вдалося
gohub_gu46_to_xml_failed	Отримати ГУ-46 у вигляді XmlDocument не вдалося
gohub_fdu92_get_xml_text_failed	Отримати ФДУ-92 у вигляді рядку не вдалося
gohub_fdu92_to_xml_failed	Отримати ФДУ-92 у вигляді XmlDocument не вдалося

## 5.17. GohubInformServicesDoc

Клас GohubInformServicesDoc використовується для представлення документів інформаційних послуг та містить наступні члени:

Id (властивість get;)	Унікальний ідентифікатор документа
Revision (властивість get;)	Номер ревізії документа
Comment (властивість get;)	Коментар документа
CreatedDate (властивість get;)	Дата створення документа
DocDate (властивість get;)	Дата документа
FileBody (властивість get;)	Тіло документа у вигляді масиву байт
FileName (властивість get;)	Найменування документа
Save (метод)	Зберегти документ в файл
SaveXml (метод)	Зберегти документ в файл формату xml

## 5.18. GohubDispatchInfo

Клас GohubDispatchInfo с використовується для представлення переліку інформації про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів(за останні 5 днів від поточної дати):

Date (поле)	Дата
Description (поле)	Повний опис
Number (поле)	Номер замовлення
Type (поле)	Тип відправки
WagOwner (поле)	Власник вагону
IsEmpty (поле)	Стан вагону порожній чи завантажений

## 5.19. Приклади використання

### а) Завантаження документа з файлу та передача в АС «Клієнт УЗ»

Продемонструємо завантаження документа з файлу та передачу його в АС «Клієнт УЗ». При цьому використовуємо електронний ключ для накладення електронно-цифрового підпису.

```
static string LoadAndSendDocument(  
    GohubConnection connection, // з'єднання з Сервером Модуля Узгодження  
    string path,                // шлях до файлу документа  
    string password)            //Пароль до ключу. Якщо не задано, документ не  
буде підписано  
{
```

```

try
{
    // Завантажити документ з файлу
    GohubDocument document = GohubDocument.Load(path);
    connection.OpenPrivateKey(password);
    Console.WriteLine("Private key info: {0}",
connection.SignerInfo);
    try
    {
        connection.SignDocument(document);
    }
    catch
    {
        throw;
    }
    finally
    {
        connection.ClosePrivateKey();
    }

    connection.SendDocument(document);
    // Документ передано успішно

    Console.WriteLine("Document is sended: id={0} revision={1}",
document.Id, document.Revision);
    // Перевірка попереджень після відправки документу
    string warning = document.Warning;
    if (!string.IsNullOrEmpty(warning))
        Console.WriteLine("Warning in document: {0}", warning);

    // Спробуємо запросити цей документ за ідентифікатором
    document = connection.QueryDocument(document.Id);
    Console.WriteLine(document.GetXmlText());
    return document.Id;
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
return null;
}

```

## б) Запит документів з АС «Клієнт УЗ»

Цей приклад демонструє техніку поступового запиту документів з АС «Клієнт УЗ» за номерами ревізій.

```

static int QueryAndSaveDocuments(
    GohubConnection connection, // з'єднання з Сервером Модуля Узгодження
    int startRevision, //номер ревізії з якої отримувати перевірки
документи
    int maxCount, //Максимальна кількість документів , що буде
запитано
    string targetFolder)//тека куди зберігати отримані документи
{
    int lastRevision = startRevision;
    int i = 0;
    var encoding = Encoding.GetEncoding(1251);
    foreach (var document in connection.QueryDocuments(lastRevision))
    {
        // Запит документа виконано успішно
        Console.WriteLine("Document received: id={0} revision={1}",
document.Id, document.Revision);
        // Вивести документ на екран
        Console.WriteLine(document.GetXmlText());
    }
}

```

```

        try
        {
            string path = string.Format("{0}\\{1}.xml", targetFolder,
document.Id);
            document.Save(path, encoding);
            Console.WriteLine("Document saved to file: path='{0}'",
path);
        }
        catch (Exception e)
        { // Помилка при збереженні документа
            Console.WriteLine(e.ToString());
        }

        // Перевірка підпису
        if (document.HasSignature)
        {
            Console.WriteLine("SignerInfo={0}\nTimeStamp={1}",
document.SignerInfo, document.TimeStamp);
        }

        // Запам'ятати ревізію останнього документа
        lastRevision = document.Revision;

        //Збільшуємо лічильник отриманих документів
        i++;
        if (i == maxCount)
            break;
    }
    // Повернення функцією номера останньої обробленої ревізії
    return lastRevision;
}

```

## в) Інші приклади використання

В інсталяційному пакеті Клієнта Модуля Узгодження представлені також інші приклади використання, з якими більш детально можна ознайомитися, заглянув безпосередньо в програмний код (тека Клієнт Модуля Узгодження\samples\ gohub.client.Test.cs). Тут ми коротко опишемо інші функції, надані у прикладах:

- Продемонстрована техніка поступового запиту документів ФДУ-92 з АС «Клієнт УЗ» за номерами ревізій

```

static ulong QueryAndSaveFdu92s(
    GohubConnection connection, // з'єднання з Сервером Модуля Узгодження
    ulong startRevision, //номер ревізії з якої отримувати перевізні
документи
    int maxCount, //Максимальна кількість документів , що буде
запитано
    string targetFolder)//тека куди зберігати отримані документи

```

- Продемонстрована техніка поступового запиту документів ЕД попереднього інформування з АС «Клієнт УЗ» за номерами ревізій

```

static ulong QueryAndSavePiPackages(
    GohubConnection connection, // з'єднання з Сервером Модуля Узгодження
    ulong startRevision, //номер ревізії з якої отримувати перевізні
документи
    int maxCount, //Максимальна кількість документів , що буде
запитано
    string targetFolder)//тека куди зберігати отримані документи

```

- Продемонстрована техніка поступового запиту пакетів попереднього інформування з АС «Клієнт УЗ» за номерами ревізій

```
static ulong QueryAndSavePiPackages(
    GohubConnection connection, // з'єднання з Сервером Модуля Узгодження
    ulong startRevision, //номер ревізії з якої отримувати перевізні
    документи
    int maxCount, //Максимальна кількість документів , що буде
    запитано
    string targetFolder)//тека куди зберігати отримані документи
```

- Продемонстрована техніка додавання документа ЕД в пакет попереднього інформування

```
static string AddEDDataToPiPackage(
    GohubConnection connection, // з'єднання з Сервером Модуля Узгодження
    string piPackageId, // ID пакета ПІ (попереднього інформування) в що
    будуть додаватися ЕД (електронні дані)
    string edataPath, // шлях до xml-файлу з ЕД
    uint edataType) // код ЕД: 190 рахунок-фактура, 320 пакувальний
    лист
```

- Продемонстрована техніка оновлення документа ЕД, що міститься на сервері АС «Клієнт УЗ»

```
static ulong UpdateEDData(
    GohubConnection connection, // з'єднання з Сервером Модуля Узгодження
    string edataId, // ID документа ЕД що буде змінюватись
    string edataPath) // шлях до xml-файлу з ЕД
```

- Продемонстрована техніка отримання переліку з інформацією про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів(за останні 5 днів від поточної дати)

```
try
{
    Console.WriteLine("Please enter ESR code station from:");
    var from_esr = Console.ReadLine();
    Console.WriteLine("Please enter ESR code station to:");
    var to_esr = Console.ReadLine();
    var list = connection.QueryDispatchInfo(from_esr, to_esr);
    if (list != null && list.Count > 0)
    {
        Console.WriteLine("list loaded.\n");
        foreach (var item in list)
        {
            Console.WriteLine(item.Description + "\n");
            Console.WriteLine(item.Number + "\n");
            Console.WriteLine(item.Date + "\n");
            Console.WriteLine(item.Type + "\n");
            Console.WriteLine(item.WagOwner + "\n");
            Console.WriteLine(item.IsEmpty + "\n");
        }
    }
    else
        Console.WriteLine("list is empty");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
```



## 6. COM/OLE бібліотека

### 6.1. Стани перевізного документа

Коди перевізного документа в *Бібліотеці* представлені переліком (*enum*) `GohubDocumentStatus`, що містить набір констант.

*Звідна таблиця кодів станів перевізного документа*

<code>gohub_document_status_unknown = 0</code>	Статус невідомий
<code>gohub_document_status_draft = 1</code>	Чернетка
<code>gohub_document_status_sending = 2</code>	Документ передається товарному касиру
<code>gohub_document_status_registered = 3</code>	Документ переданий товарному касиру
<code>gohub_document_status_reclaiming = 4</code>	Документ відкликається від товарного касира
<code>gohub_document_status_accepted = 5</code>	Вантаж прийнято до перевезення
<code>gohub_document_status_delivered = 6</code>	Вантаж прибув
<code>gohub_document_status_recieved = 7</code>	Вантаж отримано одержувачем
<code>gohub_document_status_uncredited = 8</code>	Документ розкредитовано товарним касиром
<code>gohub_document_status_recieved_draft = 9</code>	Вантаж отримано одержувачем і редагується
<code>gohub_document_status_recieved_sending = 10</code>	Вантаж отримано одержувачем і переданий товарному касиру
<code>gohub_document_status_recieved_reclaiming = 11</code>	Вантаж отримано одержувачем і відкликається від товарного касира
<code>gohub_document_status_canceled = 12</code>	Документ зіпсований товарним касиром
<code>gohub_document_status_locked = 13</code>	Документ заблокований

Більш детальний опис програмного інтерфейсу бібліотеки можна отримати з файлу `gohub.client.errors.h` (Додаток Б).

### 6.2. Ідентифікатори інтерфейсів

Звертатися до функцій бібліотеки можна з використанням технології COM/OLE. ім'я COM-компонента (ProgID): *"TMSoft.GohubClient"*, Ідентифікатор (CLSID): CF908D67-DAF6-43B0-9621-1DD417CFF3D7. Компонент надає наступні інтерфейси:

ім'я	Ідентифікатор
<code>IGohubClient</code>	ABDA6C07-5320-4F28-B995-FADE037D0A82
<code>IGohubDocument</code>	0D5D6225-8E07-46E2-8B8D-9C0966481994
<code>IGohubAttachment</code>	044603D2-574E-463C-87EC-DCD98C30F319
<code>IGohubEData</code>	685B21FA-43A7-4ACC-9A57-AB7AC332942C
<code>IGohubPiPackage</code>	62C21013-07D6-4d9d-83C4-9FA6E770B2EA
<code>IGohubPiPackageToEData</code>	88AAFC89-0426-4551-BD1C-F57F63D0C335
<code>IGohubConnection</code>	E9967B9D-1141-47BA-A5C4-573FB02DB396
<code>IGohubSignerInfo</code>	B0E1F579-5836-4E97-9340-58B092418947
<code>IGohubError</code>	F1077417-21D9-4871-84A2-9F89525E7214
<code>IGohubInformServicesDocument</code>	0af49642-b12d-4fdb-b363-a7497cc78462
<code>IGohubDispatchInfo</code>	0669AB3B-8E1B-4161-8A2E-244D5A62029A

Детальний опис продемонстровано в додатку В

## 6.3. Інтерфейс IGohubClient

Його ідентифікатор: ABDA6C07-5320-4F28-B995-FADE037D0A82

Детальний опис продемонстровано в додатку В

Методи:

Назва	Параметри	Значення, що повертається
GetLastError		IGohubError
Connect	host як String, port як Long	IGohubConnection
CreateDocument	xmlText як String	IGohubDocument
LoadDocument	path як String	IGohubDocument
LoadAttachment	typeCode як String, name як String, regNumber як String, regDate як String, validFrom як String, validTo як String, path як String	IGohubAttachment
LoadAttachmentWithUserData	typeCode як String, name як String, regNumber як String, regDate як String, validFrom як String, validTo як String, path як String, pathUserData як String	IGohubAttachment
LoadSmgsAttachment	smgsTypeCode як String, name як String, regNumber як String, regDate як String, validFrom як String, validTo як String, path як String	IGohubAttachment
LoadSmgsAttachmentWithUserData	smgsTypeCode як String, name як String, regNumber як String, regDate як String, validFrom як String, validTo як String, path як String, pathUserData як String	IGohubAttachment
MountFileKey	keyId як String, keyDir як String	Boolean
UnmountFileKey	keyId як String	Boolean
QueryMountedKeys		Long
GetMountedKeyId	index як Long	String
GetMountedKeyDir	index як Long	String
LoadEData	codeType як UInt, xmlPath як String, name як String, regNumber як String, regDate як String, validFrom як String, validTo як String, pdfPath як String	IGohubEData
LoadEDataSimple	codeType як UInt,	IGohubEData

	xmlPath як String	
LoadFdu92	ident як String, path як String	IGohubFdu92
LoadGu46	ident як String, path як String	IGohubGu46

## 6.4. Інтерфейс IGohubDocument

Його ідентифікатор: 0D5D6225-8E07-46E2-8B8D-9C0966481994

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	String
Text	String
Revision	Long
Status	Long
HasSignature	Boolean
Signer	IGohubSignerInfo
SignTime	String
AttachmentsCount	Long
MeasureEquipNum	String [get, put]
BusinessUnitNum	String [get, put]
ForeignNotAccept	Boolean
WarrantType	LONG[get, put]
OTPRString	String [get]
Warning	String [get]

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String, codePage як Long	Boolean
SaveData	Path як String, codePage як Long, epdVersion як Long	Boolean
DataText	epdVersion як Long	String
Sign	connection як IGohubConnection	Boolean
Send	connection як IGohubConnection	Boolean
GetAttachmentIdByIndex	index як Long	String
SendReceived	connection як IGohubConnection docId як String	Boolean
set_VerifiedEmptyWeightForWagon, SetVerifiedEmptyWeightForWagon	wagonIndex як LONG weight як LONG	
get_VerifiedEmptyWeightForWagon, GetVerifiedEmptyWeightForWagon	wagonIndex як LONG	LONG
GetOTPR	Path як String	Boolean

## 6.5. Інтерфейс IGohubAttachment

Його ідентифікатор: 044603D2-574E-463C-87EC-DCD98C30F319

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	String
TypeId	String
SmgsTypeId	String
Name	String
Description	String
RegNumber	String
RegDate	String
ValidFrom	String
ValidTo	String

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String	Boolean
Send	connection як IGohubConnection	Boolean
SaveUserData	Path як String	Boolean

## 6.6. Інтерфейс IGohubEData

Його ідентифікатор: 685B21FA-43A7-4ACC-9A57-AB7AC332942C

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	String
Revision	UInt64
RevisionDate	String
Version	String
DocType	UInt
Status	Int
AttachmentId	String

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String	Boolean
LoadData	Path як String	Boolean
Send	connection як IGohubConnection	Boolean
Update	connection як IGohubConnection	Boolean

## 6.7. Інтерфейс IGohubPiPackage

Його ідентифікатор: 62C21013-07D6-4d9d-83C4-9FA6E770B2EA

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	String
Revision	UInt64
RevisionDate	String
ConsignmentId	String
Status	Int
PiPackageToEDataCount	Int

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String	Boolean
PiPackageToEData	Index як Int	IGohubPiPackageToEData

## 6.8. Інтерфейс IGohubPiPackageToEData

Його ідентифікатор: 88AAFC89-0426-4551-BD1C-F57F63D0C335

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	String
EDataId	String
PiPackageId	String
Note	String
Status	Int
EDataVersion	String

## 6.9. Інтерфейс IGohubConnection

Його ідентифікатор: E9967B9D-1141-47BA-A5C4-573FB02DB396

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Signer	IGohubSignerInfo

Методи:

Назва	Параметри	Значення, що повертається
Close		
QueryDocument	docId як String	IGohubDocument
QueryNextDocument	revision як Long	IGohubDocument
OpenPrivateKey	password як String	Boolean
OpenPrivateKeyFromPath	password як String, path як String	Boolean
ReclaimDocument	docId як String	Boolean
DeleteDocument	docId як String	Boolean
QueryAttachment	attachmentId як String	IgohubAttachment
QueryAttachmentWithUserData	attachmentId як String	IgohubAttachment
DeleteAttachment	attachmentId як String	Boolean
FilterByDocumentStatus	newVal як Long	Властивості - повертає або встановлює параметр (в залежності від застосування)
FilterByDocumentNumber	newVal як String	
FilterByWagonNumber	newVal як String	
FilterByDepartureClientCode	newVal як String	
FilterByDeparturePayerCode	newVal як String	
FilterByDepartureStationCode	newVal як String	
FilterByArrivalClientCode	newVal як String	
FilterByArrivalPayerCode	newVal як String	
FilterByArrivalStationCode	newVal як String	
ClearAllFilters		Boolean
QueryAndSaveDocumentPrintableForm, SaveDocumentPrintableForm	docId як String, path як String	Boolean
QueryEData	eDataId як String	IGohubEData
QueryNextEData	revision як Long	IGohubEData
QueryPiPackage	piPackageId як String	IGohubPiPackage
QueryNextPiPackage	revision як Long	IGohubPiPackage
AddEDataToPiPackage	eData як IGohubEData piPackageId як String	IGohubPiPackage
GetMPMonths		String
QueryAndSaveOrdersForMonth	month як String, path як String	Boolean
QueryAndSaveOrdersForMonthWithRelog	month як	Boolean

in, SaveOrdersForMonthWithReLogin	String, login як String, password як String, path як String	
QueryEDataForAttachment	attachmentId як String	IGohubEData
QueryInfServsDoc	docId як Long	IGohubInformServicesDocument
QueryNextInfServsDoc	revision як Long	IGohubInformServicesDocument
QueryFdu92ByNumber	registration _esr як String, registration _num як String	GohubFdu92
QueryDispatchInfo	dispatch_esr як String, arrival_esr як String	IGohubDispatchInfo

## 6.10. Інтерфейс IGohubSignerInfo

Його ідентифікатор: B0E1F579-5836-4E97-9340-58B092418947

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Name	String
Subject	String

## 6.11. Інтерфейс IGohubError

Його ідентифікатор: F1077417-21D9-4871-84A2-9F89525E7214

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Code	Long
Title	String
Text	String

## 6.12. Інтерфейс IGohubFdu92

Його ідентифікатор: EB03BE7D-48B7-4AFD-8A94-A5012D844A17

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	String
Revision	Long
Status	Long
Text	String
HasSignature	Boolean
Signer	IGohubSignerInfo
SignTime	String

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String	Boolean
Send	connection як IGohubConnection	Boolean
Sign	connection як IGohubConnection	Boolean

## 6.13. Інтерфейс IGohubGu46

Його ідентифікатор: 78DCD121-84B7-4D41-B15C-F9F7677A3519

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	String
Revision	Long
Status	Long
Text	String
HasSignature	Boolean
Signer	IGohubSignerInfo
SignTime	String

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String	Boolean
Send	connection як IGohubConnection	Boolean
Sign	connection як IGohubConnection	Boolean

## 6.14. Інтерфейс IGohubGu45

Його ідентифікатор: 78DCD121-84B7-4D41-B15C-F9F7677A3519

Детальний опис продемонстровано в додатку В

Властивості:



Назва	Значення, що повертається
Id	String
Revision	Long
Status	Long
Text	String
HasSignature	Boolean
Signer	IGohubSignerInfo
SignTime	String

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String	Boolean

## 6.15. Інтерфейс IGohubInformServicesDocument

Його ідентифікатор: 0af49642-b12d-4fdb-b363-a7497cc78462

Детальний опис продемонстровано в додатку В

Властивості:

Назва	Значення, що повертається
Id	UInt64
Revision	UInt64
FileName	String
Comment	String
CreatedDate	String
DocDate	String

Методи:

Назва	Параметри	Значення, що повертається
Close		
Save	Path як String	Boolean
SaveXml	Path як String	Boolean

## 6.16. Інтерфейс IDispatchInfo

Його ідентифікатор: 0669AB3B-8E1B-4161-8A2E-244D5A62029A

Детальний опис продемонстровано в додатку В

Методи:

Назва	Параметри	Значення, що повертається
Close		
WagOwnerByIndex	index як Long	String
TypeByIndex	index як Long	String
DateByIndex	index як Long	String
DescriptionByIndex	index як Long	String

NumberByIndex	index як Long	String
IsEmptyByIndex	index як Long	String
Count		UInt64

## 6.17. Приклади використання

Усі наведені приклади використання Модуля Узгодження для COM/OLE бібліотеки продемонстровані мовою Visual Basic.

### а) Вивід на екран інформації про помилку

Наводимо приклад отримання інформації про помилку. Визначимо функцію `PrintError`, що виводить на екран детальну інформацію про помилку, включно з інформацією про вкладені помилки, якщо вони наявні. Ця функція стане в нагоді у наступних прикладах.

```
Sub PrintError(ByVal client As Object)
    Dim Err As Object = client.GetLastError()
    Console.WriteLine(Err.Code)
    Console.WriteLine(Err.Title)
    Console.WriteLine(Err.Text)
    Marshal.ReleaseComObject(Err)
End Sub
```

### б) Завантаження документа з файлу та передача в АС «Клієнт УЗ»

Продемонструємо завантаження документа з файлу та передачу його в АС «Клієнт УЗ». При цьому використовуємо електронний ключ для накладення електронно-цифрового підпису. Для виводу інформації про можливі помилки використовуємо функцію `PrintError`, визначену раніше в п.6.15.а.

```
Sub LoadAndSendDocument(ByRef client As Object, ByRef connection As Object, _
    ByVal path As String, _
    ByVal password As String)
    ' Завантажити документ з файлу
    Dim document As Object = client.LoadDocument(path)
    If (document Is Nothing) Then
        ' Помилка при завантаженні документа з файлу
        PrintError(client)
        Return
    End If
    ' Документ завантажено успішно
    Console.WriteLine("Document is loaded")

    ' Підписати документ
    If Not connection.OpenPrivateKey(password) Then
        ' Не вдалося отримати доступ до закритого ключу
        PrintError(client)
        CloseAndRelease(document)
        Return
    End If
    Console.WriteLine("Signing document . . .")
    Console.WriteLine("Private key info: name={0}, Subject={1}",
        connection.Signer.Name, connection.Signer.Subject)

    If Not document.Sign(connection) Then
        ' Не вдалося отримати доступ до закритого ключу
        PrintError(client)
        CloseAndRelease(document)
        Return
    End If
```

```

' Передати документ в АС "Клієнт УЗ" засобами Модуля Узгодження
If Not document.Send(connection) Then
    ' Помилка при передачі документа
    PrintError(client)
    CloseAndRelease(document)
    Return
End If

' Документ передано успішно
Console.WriteLine("Document is sended: id={0} revision={1}", document.Id,
document.Revision)

'Перевірка попередження після відправки документа
Dim warning As String
warning = documnet.Warning
If Not String.IsNullOrEmpty(warning) Then
    Console.WriteLine("Warning: {0}", warning)
End If

' Зберігаємо ідентифікатор документа для подальшого використання
Dim id As String = document.Id
CloseAndRelease(document)

' Спробуємо запросити цей документ за ідентифікатором
document = connection.QueryDocument(id)
If (document Is Nothing) Then
    ' Помилка при отриманні документа
    PrintError(client)
    Return
End If

Console.WriteLine("Document is queried.")
Console.WriteLine(document.Text)

CloseAndRelease(document)
End Sub

```

#### в) Запит документів з АС «Клієнт УЗ»

Цей приклад демонструє техніку поступового запиту документів їх АС «Клієнт УЗ» за номерами ревізій. Окрім того показано зміни кодової сторінки документа та збереження запитаних документів у файли. Окрім того, виконується перевірка електронно-цифрового підпису. Для виводу інформації про помилки використовується функція `PrintError`, визначена раніше в п.6.15.а

```

Function QueryAndSaveDocuments(ByRef client As Object, ByRef connection As
Object, _
                                ByVal startRevision As Integer, _
                                ByVal maxCount As Integer, _
                                ByVal targetFolder As String _
                                ) As Integer
    Dim lastRevision As Integer = startRevision
    Dim document As Object
    For count As Integer = 0 To maxCount
        ' Запит наступного документа за ревізією
        document = connection.QueryNextDocument(lastRevision)
        If (document Is Nothing) Then
            PrintError(client)
            Exit For
        End If
        ' Запит документа виконано успішно
    
```

```

' Вивести документ на екран
Console.WriteLine("Document queried: Id={0}, Rev={1}", document.Id,
document.Revision)

Dim path As String = targetFolder + "\" + document.Id + ".xml"
If (document.Save(path, 866)) Then
    Console.WriteLine("Document saved to file: path={0}", path)
Else
    'Помилка при збереженні документа
    PrintError(client)
End If

' Перевірка підпису
If (document.HasSignature) Then
    Console.WriteLine("Signer: Name={0}, Subject={1}. TimeStamp={2}",
document.Signer.Name, document.Signer.Subject, document.SignTime)
End If
' Запам'ятати ревізію останнього документа
lastRevision = document.Revision
CloseAndRelease(document)

Next
Return lastRevision
End Function

```

## г) Інші приклади використання

В інсталяційному пакеті Клієнта Модуля Узгодження представлені також інші приклади використання, з якими більш детально можна ознайомитися, заглянув безпосередньо в програмний код (тека Клієнт Модуля Узгодження\samples\ Gohub.Client.COM.Test.VB). Тут ми коротко опишемо інші функції, надані у прикладах:

- Продемонстрована техніка поступового запиту документів ФДУ-92 з АС «Клієнт УЗ» за номерами ревізій

```

Function QueryAndSaveFdu92s(ByRef client As Object, ByRef connection As
Object, _
                                ByVal startRevision As Integer, _
                                ByVal maxCount As Integer, _
                                ByVal targetFolder As String _
                                ) As Integer

```

- Продемонстрована техніка поступового запиту документів ЕД попереднього інформування з АС «Клієнт УЗ» за номерами ревізій

```

Function QueryAndSaveEDatas(ByRef client As Object, ByRef connection As
Object, _
                                ByVal startRevision As ULong, _
                                ByVal maxCount As Integer, _
                                ByVal targetFolder As String _
                                ) As ULong

```

- Продемонстрована техніка поступового запиту пакетів попереднього інформування з АС «Клієнт УЗ» за номерами ревізій

```

Function QueryAndSavePiPackages(ByRef client As Object, ByRef connection As
Object, _
                                ByVal startRevision As ULong, _
                                ByVal maxCount As Integer, _
                                ByVal targetFolder As String _
                                ) As ULong

```

- Продемонстрована техніка додавання документа ЕД в пакет попереднього інформування

```
Function AddEDDataToPiPackage(ByRef client As Object, ByRef connection As
Object, _
                                ByVal piPackageId As String, _
                                ByVal edataPath As String, _
                                ByVal edataType As Integer _
                                ) As String
```

- Продемонстрована техніка оновлення документа ЕД, що міститься на сервері АС «Клієнт УЗ»

```
Function UpdateEDData(ByRef client As Object, ByRef connection As Object, _
                    ByVal edataId As String, _
                    ByVal edataPath As String _
                    ) As ULong
```

- Продемонстрована техніка отримання переліку з інформацією про замовлення на погодження перевезення за даними календаря планування перевезень зернових вантажів(за останні 5 днів від поточної дати)

```
Try
    Console.WriteLine(" Please enter ESR number of start station: ")
    Dim start_esr = Console.ReadLine()
    Console.WriteLine(" Please enter ESR number of end station: ")
    Dim end_ser = Console.ReadLine()
    Dim doc = connection.QueryDispatchInfo(start_esr, end_ser)
    PrintError(client.GetLastError())
    If doc IsNot Nothing Then
        Dim count = doc.Count()
        If count > 0 Then
            Dim i = 0
            While i < count
                Console.WriteLine(doc.DescriptionByIndex(i))
                Console.WriteLine(doc.NumberByIndex(i))
                Console.WriteLine(doc.DateByIndex(i))
                Console.WriteLine(doc.TypeByIndex(i))
                Console.WriteLine(doc.WagOwnerByIndex(i))
                Console.WriteLine(doc.IsEmptyByIndex(i))
                i += 1
            End While
        End If
        doc.Close()
    End If
Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.ReadKey()
```

## ДОДАТОК А. Файл *gohub.client.h*

```
#ifndef GOHUB_CLIENT_H_
#define GOHUB_CLIENT_H_

#include "gohub.client.errors.h"

#ifdef __cplusplus
extern "C" {
#endif

//////////
// Загальні типи //
//////////
typedef int GohubBool;
typedef unsigned short GohubWChar;
typedef struct GohubConnection GohubConnection;
typedef struct GohubDocument GohubDocument;
typedef struct GohubAttachment GohubAttachment;
typedef struct GohubEData GohubEData;
typedef struct GohubPiPackageToEData GohubPiPackageToEData;
typedef struct GohubPiPackage GohubPiPackage;
typedef struct GohubFdu92 GohubFdu92;
typedef struct GohubGu46 GohubGu46;
typedef struct GohubGu45 GohubGu45;
typedef struct GohubGu27 GohubGu27;
typedef struct GohubInformServicesDoc GohubInformServicesDoc;

//////////
// Статуси перевізного документа //
//////////
typedef enum GohubDocumentStatus
{
    gohub_document_status_unknown,
    gohub_document_status_draft,
    gohub_document_status_sending,
    gohub_document_status_registered,
    gohub_document_status_reclaiming,
    gohub_document_status_accepted,
    gohub_document_status_delivered,
    gohub_document_status_recieved,
    gohub_document_status_uncredited,
    gohub_document_status_recieved_draft,
    gohub_document_status_recieved_sending,
    gohub_document_status_recieved_reclaiming,
    gohub_document_status_canceled,
    gohub_document_status_locked,
} GohubDocumentStatus;

//////////
// Статуси ГУ-27 //
//////////
typedef enum GohubGu27Status
{
    gohub_gu27_status_unknown,
    gohub_gu27_status_draft,
    gohub_gu27_status_sending,
    gohub_gu27_status_registered,
    gohub_gu27_status_reclaiming,
    gohub_gu27_status_accepted,
    gohub_gu27_status_delivered,
    gohub_gu27_status_recieved,
    gohub_gu27_status_uncredited,
    gohub_gu27_status_recieved_draft,
    gohub_gu27_status_recieved_sending,
```

```

    gohub_gu27_status_recieved_reclaiming,
    gohub_gu27_status_canceled,
    gohub_gu27_status_locked,
} GohubGu27Status;

////////////////////////////////////
// Робота з кодовими сторінками //
////////////////////////////////////
int gohub_codepage();
int gohub_set_codepage(int codepage);
int gohub_encoding_codepage(const char* encodingName);
int gohub_encoding_codepage_w(const GohubWChar* encodingName);

////////////////////////////////////
// Підключення до Модулю Узгодження //
////////////////////////////////////
GohubConnection* gohub_connect(const char* host, int port);
GohubConnection* gohub_connect_w(const GohubWChar* host, int port);
GohubBool gohub_disconnect(GohubConnection* connection);

////////////////////////////////////
// Робота з документами //
////////////////////////////////////
GohubDocument* gohub_load_document(const char* path);
GohubDocument* gohub_load_document_w(const GohubWChar* path);
GohubDocument* gohub_create_document(const char* content);
GohubDocument* gohub_create_document_w(const GohubWChar* content);
GohubDocument* gohub_query_document(GohubConnection* connection, const char*
documentId);
GohubDocument* gohub_query_document_w(GohubConnection* connection, const
GohubWChar* documentId);
GohubDocument* gohub_query_next_document(GohubConnection* connection, int
lastRevision);
const char* gohub_document_id(GohubDocument* document);
const GohubWChar* gohub_document_id_w(GohubDocument* document);
int gohub_document_revision(GohubDocument* document);
const char* gohub_document_text(GohubDocument* document);
const GohubWChar* gohub_document_text_w(GohubDocument* document);
const char* gohub_document_data_text(GohubDocument* document, int epdVersion);
const GohubWChar* gohub_document_data_text_w(GohubDocument* document, int
epdVersion);
int gohub_document_size(GohubDocument* document);
const char* gohub_document_measure_equip_num(GohubDocument* document);
const GohubWChar* gohub_document_measure_equip_num_w(GohubDocument* document);
const char* gohub_document_business_unit_num(GohubDocument* document);
const GohubWChar* gohub_document_business_unit_num_w(GohubDocument* document);
int gohub_document_get_verified_empty_weight_for_wagon(GohubDocument* document,
int wagonIndex);
bool gohub_document_get_foreign_not_accept(GohubDocument* document);
int gohub_document_warrant_type(GohubDocument* document);
GohubBool gohub_document_set_measure_equip_num(GohubDocument* document, const
char* val);
GohubBool gohub_document_set_measure_equip_num_w(GohubDocument* document, const
GohubWChar* val);
GohubBool gohub_document_set_business_unit_num(GohubDocument* document, const
char* val);
GohubBool gohub_document_set_business_unit_num_w(GohubDocument* document, const
GohubWChar* val);
GohubBool gohub_document_set_warrant_type(GohubDocument* document, int val);
GohubBool gohub_document_set_verified_empty_weight_for_wagon(GohubDocument*
document, int wagonIndex, int val);
GohubBool gohub_send_document(GohubConnection* connection, GohubDocument*
document);
GohubBool gohub_send_received_document(GohubConnection* connection,
GohubDocument* document, const char* documentId);

```

```

GohubBool gohub_send_received_document_w(GohubConnection* connection,
GohubDocument* document, const GohubWChar* documentId);
GohubBool gohub_save_document(GohubDocument* document, const char* path, int
codePage);
GohubBool gohub_save_document_w(GohubDocument* document, const GohubWChar* path,
int codePage);
GohubBool gohub_save_document_data(GohubDocument* document, const char* path, int
codePage, int epdVersion);
GohubBool gohub_save_document_data_w(GohubDocument* document, const GohubWChar*
path, int codePage, int epdVersion);
GohubBool gohub_close_document(GohubDocument* document);
GohubBool gohub_reclaim_document(GohubConnection* connection, const char*
documentId);
GohubBool gohub_reclaim_document_w(GohubConnection* connection, const GohubWChar*
documentId);
GohubBool gohub_delete_document(GohubConnection* connection, const char*
documentId);
GohubBool gohub_delete_document_w(GohubConnection* connection, const GohubWChar*
documentId);
GohubDocumentStatus gohub_document_status(GohubDocument* document);
GohubBool gohub_document_get_otpr(GohubDocument* document, const char* path);
GohubBool gohub_document_get_otpr_w(GohubDocument* document, const GohubWChar*
path);
const char* gohub_document_get_otpr_string(GohubDocument* document);
const GohubWChar* gohub_document_get_otpr_string_w(GohubDocument* document);
const char* gohub_document_warning(GohubDocument* document);
const GohubWChar* gohub_document_warning_w(GohubDocument* document);

////////////////////////////////////
// Робота з фільтрами запиту документів //
////////////////////////////////////
// Накладення фільтрів впливає на результат роботи функції
gohub_query_next_document
GohubBool gohub_clear_all_filters(GohubConnection* connection);
GohubBool gohub_set_filter_by_document_status(GohubConnection* connection, int
documentStatusCode);
GohubBool gohub_set_filter_by_document_number(GohubConnection* connection, const
char* documentNumber);
GohubBool gohub_set_filter_by_document_number_w(GohubConnection* connection,
const GohubWChar* documentNumber);
GohubBool gohub_set_filter_by_wagon_number(GohubConnection* connection, const
char* wagonNumber);
GohubBool gohub_set_filter_by_wagon_number_w(GohubConnection* connection, const
GohubWChar* wagonNumber);
GohubBool gohub_set_filter_by_departure_client(GohubConnection* connection, const
char* clientCode);
GohubBool gohub_set_filter_by_departure_client_w(GohubConnection* connection,
const GohubWChar* clientCode);
GohubBool gohub_set_filter_by_departure_payer(GohubConnection* connection, const
char* payerCode);
GohubBool gohub_set_filter_by_departure_payer_w(GohubConnection* connection,
const GohubWChar* payerCode);
GohubBool gohub_set_filter_by_departure_station(GohubConnection* connection,
const char* stationCode);
GohubBool gohub_set_filter_by_departure_station_w(GohubConnection* connection,
const GohubWChar* stationCode);
GohubBool gohub_set_filter_by_arrival_client(GohubConnection* connection, const
char* clientCode);
GohubBool gohub_set_filter_by_arrival_client_w(GohubConnection* connection, const
GohubWChar* clientCode);
GohubBool gohub_set_filter_by_arrival_payer(GohubConnection* connection, const
char* payerCode);
GohubBool gohub_set_filter_by_arrival_payer_w(GohubConnection* connection, const
GohubWChar* payerCode);

```



```

GohubBool gohub_set_filter_by_arrival_station(GohubConnection* connection, const
char* stationCode);
GohubBool gohub_set_filter_by_arrival_station_w(GohubConnection* connection,
const GohubWChar* stationCode);

GohubDocumentStatus gohub_get_filter_by_document_status(GohubConnection*
connection);
const char* gohub_get_filter_by_document_number(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_document_number_w(GohubConnection*
connection);
const char* gohub_get_filter_by_wagon_number(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_wagon_number_w(GohubConnection*
connection);
const char* gohub_get_filter_by_departure_client(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_departure_client_w(GohubConnection*
connection);
const char* gohub_get_filter_by_departure_payer(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_departure_payer_w(GohubConnection*
connection);
const char* gohub_get_filter_by_departure_station(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_departure_station_w(GohubConnection*
connection);
const char* gohub_get_filter_by_arrival_client(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_arrival_client_w(GohubConnection*
connection);
const char* gohub_get_filter_by_arrival_payer(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_arrival_payer_w(GohubConnection*
connection);
const char* gohub_get_filter_by_arrival_station(GohubConnection* connection);
const GohubWChar* gohub_get_filter_by_arrival_station_w(GohubConnection*
connection);

////////////////////////////////////////
// Запит друкованих форм документів //
////////////////////////////////////////
GohubBool gohub_query_and_save_document_printable_form(
    GohubConnection* connection,
    const char* documentId,
    const char* path);

GohubBool gohub_query_and_save_document_printable_form_w(
    GohubConnection* connection,
    const GohubWChar* documentId,
    const GohubWChar* path);

GohubBool gohub_query_and_save_fdu92_printable_form(
    GohubConnection* connection,
    const char* fdu92Id,
    const char* path);

GohubBool gohub_query_and_save_fdu92_printable_form_w(
    GohubConnection* connection,
    const GohubWChar* fdu92Id,
    const GohubWChar* path);

GohubBool gohub_query_and_save_gu46_printable_form(
    GohubConnection* connection,
    const char* gu46Id,
    const char* path);

GohubBool gohub_query_and_save_gu46_printable_form_w(
    GohubConnection* connection,
    const GohubWChar* gu46Id,
    const GohubWChar* path);

```

```

GohubBool gohub_query_and_save_gu45_printable_form(
    GohubConnection* connection,
    const char* gu45Id,
    const char* path);

GohubBool gohub_query_and_save_gu45_printable_form_w(
    GohubConnection* connection,
    const GohubWChar* gu45Id,
    const GohubWChar* path);

GohubBool gohub_query_and_save_gu27_printable_form(
    GohubConnection* connection,
    const char* gu27Id,
    const char* path);

GohubBool gohub_query_and_save_gu27_printable_form_w(
    GohubConnection* connection,
    const GohubWChar* gu27Id,
    const GohubWChar* path);

////////////////////////////////////
// Робота зі супровідними документами //
////////////////////////////////////
GohubAttachment* gohub_load_attachment(
    const char* typeCode,
    const char* name,
    const char* regNumber,
    const char* regDate,
    const char* validFrom,
    const char* validTo,
    const char* path);
GohubAttachment* gohub_load_attachment_w(
    const GohubWChar* typeCode,
    const GohubWChar* name,
    const GohubWChar* regNumber,
    const GohubWChar* regDate,
    const GohubWChar* validFrom,
    const GohubWChar* validTo,
    const GohubWChar* path);
GohubAttachment* gohub_load_smgs_attachment(
    const char* smgsTypeCode,
    const char* name,
    const char* regNumber,
    const char* regDate,
    const char* validFrom,
    const char* validTo,
    const char* path);
GohubAttachment* gohub_load_smgs_attachment_w(
    const GohubWChar* smgsTypeCode,
    const GohubWChar* name,
    const GohubWChar* regNumber,
    const GohubWChar* regDate,
    const GohubWChar* validFrom,
    const GohubWChar* validTo,
    const GohubWChar* path);
GohubAttachment* gohub_load_attachment_with_user_data(
    const char* typeCode,
    const char* name,
    const char* regNumber,
    const char* regDate,
    const char* validFrom,
    const char* validTo,
    const char* pathPdf,
    const char* pathUserData);
GohubAttachment* gohub_load_attachment_with_user_data_w(

```

```

    const GohubWChar* typeCode,
    const GohubWChar* name,
    const GohubWChar* regNumber,
    const GohubWChar* regDate,
    const GohubWChar* validFrom,
    const GohubWChar* validTo,
    const GohubWChar* pathPdf,
    const GohubWChar* pathUserData);
GohubAttachment* gohub_load_smgs_attachment_with_user_data(
    const char* smgsTypeCode,
    const char* name,
    const char* regNumber,
    const char* regDate,
    const char* validFrom,
    const char* validTo,
    const char* pathPdf,
    const char* pathUserData);
GohubAttachment* gohub_load_smgs_attachment_with_user_data_w(
    const GohubWChar* smgsTypeCode,
    const GohubWChar* name,
    const GohubWChar* regNumber,
    const GohubWChar* regDate,
    const GohubWChar* validFrom,
    const GohubWChar* validTo,
    const GohubWChar* pathPdf,
    const GohubWChar* pathUserData);
GohubBool gohub_send_attachment(GohubConnection* connection, GohubAttachment*
attachment);
GohubAttachment* gohub_query_attachment(GohubConnection* connection, const char*
attachmentId);
GohubAttachment* gohub_query_attachment_w(GohubConnection* connection, const
GohubWChar* attachmentId);
GohubAttachment* gohub_query_attachment_with_user_data(GohubConnection*
connection, const char* attachmentId);
GohubAttachment* gohub_query_attachment_with_user_data_w(GohubConnection*
connection, const GohubWChar* attachmentId);
GohubBool gohub_save_attachment(GohubAttachment* attachment, const char* path);
GohubBool gohub_save_attachment_w(GohubAttachment* attachment, const GohubWChar*
path);
GohubBool gohub_save_attachment_with_user_data(GohubAttachment* attachment, const
char* path);
GohubBool gohub_save_attachment_with_user_data_w(GohubAttachment* attachment,
const GohubWChar* path);
GohubBool gohub_delete_attachment(GohubConnection* connection, const char*
attachmentId);
GohubBool gohub_delete_attachment_w(GohubConnection* connection, const
GohubWChar* attachmentId);
GohubBool gohub_close_attachment(GohubAttachment* attachment);
const char* gohub_attachment_id(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_id_w(GohubAttachment* attachment);
const char* gohub_attachment_description(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_description_w(GohubAttachment* attachment);
const char* gohub_attachment_type_code(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_type_code_w(GohubAttachment* attachment);
const char* gohub_attachment_smgs_type_code(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_smgs_type_code_w(GohubAttachment* attachment);
const char* gohub_attachment_name(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_name_w(GohubAttachment* attachment);
const char* gohub_attachment_reg_number(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_reg_number_w(GohubAttachment* attachment);
const char* gohub_attachment_reg_date(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_reg_date_w(GohubAttachment* attachment);
const char* gohub_attachment_valid_from(GohubAttachment* attachment);
const GohubWChar* gohub_attachment_valid_from_w(GohubAttachment* attachment);
const char* gohub_attachment_valid_to(GohubAttachment* attachment);

```

```

const GohubWChar* gohub_attachment_valid_to_w(GohubAttachment* attachment);
int gohub_attachment_count(GohubDocument* document);
const char* gohub_attachment_id_by_index(GohubDocument* document, int
attachmentIndex);
const GohubWChar* gohub_attachment_id_by_index_w(GohubDocument* document, int
attachmentIndex);

////////////////////////////////////
// Робота з електронними даними попереднього інформування //
////////////////////////////////////
GohubEData* gohub_load_edata(
    unsigned int attachmentSmgsTypeCode,
    const char* xmlPath,
    const char* name,
    const char* regNumber,
    const char* regDate,
    const char* validFrom,
    const char* validTo,
    const char* pdfPath);
GohubEData* gohub_load_edata_w(
    unsigned int attachmentSmgsTypeCode,
    const GohubWChar* xmlPath,
    const GohubWChar* name,
    const GohubWChar* regNumber,
    const GohubWChar* regDate,
    const GohubWChar* validFrom,
    const GohubWChar* validTo,
    const GohubWChar* pdfPath);
GohubEData* gohub_load_edata_simple(
    unsigned int attachmentSmgsTypeCode,
    const char* xmlPath);
GohubEData* gohub_load_edata_simple_w(
    unsigned int attachmentSmgsTypeCode,
    const GohubWChar* xmlPath);
GohubBool gohub_edata_load_data(GohubEData* edata, const char* xmlPath);
GohubBool gohub_edata_load_data_w(GohubEData* edata, const GohubWChar* xmlPath);
GohubBool gohub_update_edata(GohubConnection* connection, GohubEData* eData);
GohubBool gohub_send_edata(GohubConnection* connection, GohubEData* eData);
GohubEData* gohub_query_edata(GohubConnection* connection, const char* eDataId);
GohubEData* gohub_query_edata_w(GohubConnection* connection, const GohubWChar*
eDataId);
GohubEData* gohub_query_edata_for_attachment(GohubConnection* connection, const
char* attachmentId);
GohubEData* gohub_query_edata_for_attachment_w(GohubConnection* connection, const
GohubWChar* attachmentId);
GohubEData* gohub_query_next_edata(GohubConnection* connection, unsigned __int64
lastRevision);
GohubBool gohub_save_edata(GohubEData* eData, const char* path);
GohubBool gohub_save_edata_w(GohubEData* eData, const GohubWChar* path);
GohubBool gohub_close_edata(GohubEData* eData);
const char* gohub_edata_id(GohubEData* eData);
const GohubWChar* gohub_edata_id_w(GohubEData* eData);
unsigned __int64 gohub_edata_revision(GohubEData* eData);
const char* gohub_edata_revision_date(GohubEData* eData);
const GohubWChar* gohub_edata_revision_date_w(GohubEData* eData);
unsigned int gohub_edata_doc_type(GohubEData* eData);
int gohub_edata_status(GohubEData* eData);
const char* gohub_edata_version(GohubEData* eData);
const GohubWChar* gohub_edata_version_w(GohubEData* eData);
const char* gohub_edata_attachment_id(GohubEData* eData);
const GohubWChar* gohub_edata_attachment_id_w(GohubEData* eData);

////////////////////////////////////
// Робота з пакетами попереднього інформування //
////////////////////////////////////

```

```

GohubPiPackage* gohub_query_pi_package(GohubConnection* connection, const char*
piPackageId);
GohubPiPackage* gohub_query_pi_package_w(GohubConnection* connection, const
GohubWChar* piPackageId);
GohubPiPackage* gohub_query_next_pi_package(GohubConnection* connection, unsigned
__int64 lastRevision);
GohubBool gohub_save_pi_package(GohubPiPackage* piPackage, const char* path);
GohubBool gohub_save_pi_package_w(GohubPiPackage* piPackage, const GohubWChar*
path);
GohubBool gohub_close_pi_package(GohubPiPackage* piPackage);
const char* gohub_pi_package_id(GohubPiPackage* piPackage);
const GohubWChar* gohub_pi_package_id_w(GohubPiPackage* piPackage);
unsigned __int64 gohub_pi_package_revision(GohubPiPackage* piPackage);
const char* gohub_pi_package_revision_date(GohubPiPackage* piPackage);
const GohubWChar* gohub_pi_package_revision_date_w(GohubPiPackage* piPackage);
const char* gohub_pi_package_consignment_id(GohubPiPackage* piPackage);
const GohubWChar* gohub_pi_package_consignment_id_w(GohubPiPackage* piPackage);
int gohub_pi_package_status(GohubPiPackage* piPackage);
int gohub_pi_package_pipacktoed_count(GohubPiPackage* piPackage);
GohubPiPackageToEData* gohub_pi_package_pipacktoed(GohubPiPackage* piPackage, int
index);

const char* gohub_pipacktoed_id(GohubPiPackageToEData* piPackageToEData);
const GohubWChar* gohub_pipacktoed_id_w(GohubPiPackageToEData* piPackageToEData);
const char* gohub_pipacktoed_edata_id(GohubPiPackageToEData* piPackageToEData);
const GohubWChar* gohub_pipacktoed_edata_id_w(GohubPiPackageToEData*
piPackageToEData);
const char* gohub_pipacktoed_pi_package_id(GohubPiPackageToEData*
piPackageToEData);
const GohubWChar* gohub_pipacktoed_pi_package_id_w(GohubPiPackageToEData*
piPackageToEData);
const char* gohub_pipacktoed_note(GohubPiPackageToEData* piPackageToEData);
const GohubWChar* gohub_pipacktoed_note_w(GohubPiPackageToEData*
piPackageToEData);
int gohub_pipacktoed_status(GohubPiPackageToEData* piPackageToEData);
const char* gohub_pipacktoed_edata_version(GohubPiPackageToEData*
piPackageToEData);
const GohubWChar* gohub_pipacktoed_edata_version_w(GohubPiPackageToEData*
piPackageToEData);

GohubPiPackage* gohub_add_edata_to_pi_package(GohubConnection* connection,
GohubEData* edata, const char* piPackageId);
GohubPiPackage* gohub_add_edata_to_pi_package_w(GohubConnection* connection,
GohubEData* edata, const GohubWChar* piPackageId);

////////////////////////////////////
// Статуси електронних ФДУ-92//
////////////////////////////////////
typedef enum GohubFdu92Status
{
    gohub_fdu92_status_unknown = -1,
    gohub_fdu92_status_approving = 0,
    gohub_fdu92_status_approving_modified = 1,
    gohub_fdu92_status_confirmed = 2,
    gohub_fdu92_status_canceled = 3,
    gohub_fdu92_status_agreed_noted_sending = 4,
    gohub_fdu92_status_agreed = 5,
    gohub_fdu92_status_agreed_noted = 6,
    gohub_fdu92_status_expired = 7,
    gohub_fdu92_status_agreed_sending = 8,
    gohub_fdu92_status_confirmed_paper = 9,
    gohub_fdu92_status_rejecting = 10,
    gohub_fdu92_status_rejected = 11,
    gohub_fdu92_status_paper = 60,
} GohubFdu92Status;

```

```

////////////////////////////////////
// Робота з електронними ФДУ-92 //
////////////////////////////////////
GohubFdu92* gohub_load_fdu92(const char* id, const char* path);
GohubFdu92* gohub_load_fdu92_w(const GohubWChar* id, const GohubWChar* path);
GohubFdu92* gohub_create_fdu92(const char* id, const char* content);
GohubFdu92* gohub_create_fdu92_w(const char* id, const GohubWChar* content);
GohubFdu92* gohub_query_fdu92(GohubConnection* connection, const char* fdu92Id);
GohubFdu92* gohub_query_fdu92_w(GohubConnection* connection, const GohubWChar*
fdu92Id);
GohubFdu92* gohub_query_next_fdu92(GohubConnection* connection, int
lastRevision);
GohubFdu92* gohub_query_fdu92_by_number(GohubConnection* connection, const char*
registration_esr, const char* registration_num);
GohubFdu92* gohub_query_fdu92_by_number_w(GohubConnection* connection, const
GohubWChar* registration_esr, const GohubWChar* registration_num);
GohubBool gohub_save_fdu92(GohubFdu92* fdu92, const char* path, int codePage);
GohubBool gohub_save_fdu92_w(GohubFdu92* fdu92, const GohubWChar* path, int
codePage);
GohubBool gohub_close_fdu92(GohubFdu92* fdu92);
GohubBool gohub_send_fdu92(GohubConnection* connection, GohubFdu92* document);
GohubBool gohub_reject_fdu92(GohubConnection* connection, const char* id);
GohubBool gohub_reject_fdu92_w(GohubConnection* connection, const GohubWChar*
id);

const char* gohub_fdu92_id(GohubFdu92* fdu92);
const GohubWChar* gohub_fdu92_id_w(GohubFdu92* fdu92);
int gohub_fdu92_revision(GohubFdu92* fdu92);
GohubFdu92Status gohub_fdu92_status(GohubFdu92* fdu92);
const char* gohub_fdu92_text(GohubFdu92* fdu92);
const GohubWChar* gohub_fdu92_text_w(GohubFdu92* fdu92);
int gohub_fdu92_size(GohubFdu92* fdu92);
const char* gohub_fdu92_signer_info(GohubFdu92* fdu92);
const GohubWChar* gohub_fdu92_signer_info_w(GohubFdu92* fdu92);
const char* gohub_fdu92_sign_time(GohubFdu92* fdu92);
const GohubWChar* gohub_fdu92_sign_time_w(GohubFdu92* fdu92);
GohubBool gohub_fdu92_has_signature(GohubFdu92* fdu92);
const GohubWChar* gohub_fdu92_signer_name_w(GohubFdu92* fdu92);

////////////////////////////////////
// Статуси електронних ГУ-46 //
////////////////////////////////////
typedef enum GohubGu46Status
{
    gohub_gu46_status_unknown = -1,
    gohub_gu46_status_approving = 0,
    gohub_gu46_status_approving_modified = 1,
    gohub_gu46_status_confirmed = 2,
    gohub_gu46_status_canceled = 3,
    gohub_gu46_status_agreed_noted_sending = 4,
    gohub_gu46_status_agreed = 5,
    gohub_gu46_status_agreed_noted = 6,
    gohub_gu46_status_expired = 7,
    gohub_gu46_status_agreed_sending = 8,
    gohub_gu46_status_confirmed_paper = 9,
    gohub_gu46_status_rejecting = 10,
    gohub_gu46_status_rejected = 11,
    gohub_gu46_status_paper = 60,
} GohubGu46Status;

////////////////////////////////////
// Робота з електронними ГУ-46 //
////////////////////////////////////
GohubGu46* gohub_load_gu46(const char* id, const char* path);
GohubGu46* gohub_load_gu46_w(const GohubWChar* id, const GohubWChar* path);

```

```

GohubGu46* gohub_create_gu46(const char* id, const char* content);
GohubGu46* gohub_create_gu46_w(const GohubWChar* id, const GohubWChar* content);
GohubGu46* gohub_query_gu46(GohubConnection* connection, const char* gu46Id);
GohubGu46* gohub_query_gu46_w(GohubConnection* connection, const GohubWChar*
gu46Id);
GohubGu46* gohub_query_next_gu46(GohubConnection* connection, int lastRevision);
GohubBool gohub_save_gu46(GohubGu46* gu46, const char* path, int codePage);
GohubBool gohub_save_gu46_w(GohubGu46* gu46, const GohubWChar* path, int
codePage);
GohubBool gohub_close_gu46(GohubGu46* gu46);
GohubBool gohub_send_gu46(GohubConnection* connection, GohubGu46* document);
GohubBool gohub_reject_gu46(GohubConnection* connection, const char* id);
GohubBool gohub_reject_gu46_w(GohubConnection* connection, const GohubWChar* id);

const char* gohub_gu46_id(GohubGu46* gu46);
const GohubWChar* gohub_gu46_id_w(GohubGu46* gu46);
int gohub_gu46_revision(GohubGu46* gu46);
GohubGu46Status gohub_gu46_status(GohubGu46* gu46);
const char* gohub_gu46_text(GohubGu46* gu46);
const GohubWChar* gohub_gu46_text_w(GohubGu46* gu46);
int gohub_gu46_size(GohubGu46* gu46);
const char* gohub_gu46_signer_info(GohubGu46* gu46);
const GohubWChar* gohub_gu46_signer_info_w(GohubGu46* gu46);
const char* gohub_gu46_sign_time(GohubGu46* gu46);
const GohubWChar* gohub_gu46_sign_time_w(GohubGu46* gu46);
GohubBool gohub_gu46_has_signature(GohubGu46* gu46);
const GohubWChar* gohub_gu46_signer_name_w(GohubGu46* gu46);

//////////
// Статуси електронних ГУ-45 //
//////////
typedef enum GohubGu45Status
{
    gohub_gu45_status_unknown = -1,
    gohub_gu45_status_confirmed = 2,
    gohub_gu45_status_canceled = 3,
    gohub_gu45_status_confirmed_paper = 9,
    gohub_gu45_status_paper = 60,
} GohubGu45Status;
//////////
// Робота з електронними ГУ-45 //
//////////
GohubGu45* gohub_query_gu45(GohubConnection* connection, const char* gu45Id);
GohubGu45* gohub_query_gu45_w(GohubConnection* connection, const GohubWChar*
gu45Id);
GohubGu45* gohub_query_next_gu45(GohubConnection* connection, int lastRevision);
GohubBool gohub_save_gu45(GohubGu45* gu45, const char* path, int codePage);
GohubBool gohub_save_gu45_w(GohubGu45* gu45, const GohubWChar* path, int
codePage);
GohubBool gohub_close_gu45(GohubGu45* gu45);

const char* gohub_gu45_id(GohubGu45* gu45);
const GohubWChar* gohub_gu45_id_w(GohubGu45* gu45);
int gohub_gu45_revision(GohubGu45* gu45);
GohubGu45Status gohub_gu45_status(GohubGu45* gu45);
const char* gohub_gu45_text(GohubGu45* gu45);
const GohubWChar* gohub_gu45_text_w(GohubGu45* gu45);
int gohub_gu45_size(GohubGu45* gu45);
const char* gohub_gu45_signer_info(GohubGu45* gu45);
const GohubWChar* gohub_gu45_signer_info_w(GohubGu45* gu45);
const char* gohub_gu45_sign_time(GohubGu45* gu45);
const GohubWChar* gohub_gu45_sign_time_w(GohubGu45* gu45);
GohubBool gohub_gu45_has_signature(GohubGu45* gu45);
const GohubWChar* gohub_gu45_signer_name_w(GohubGu45* gu45);

```

```

////////////////////////////////////
// Перевірка електронно-цифрового підпису //
////////////////////////////////////
GohubBool gohub_document_has_signature(GohubDocument* document);
GohubBool gohub_document_check_signature(GohubDocument* document);
const char* gohub_document_signer_name(GohubDocument* document);
const GohubWChar* gohub_document_signer_name_w(GohubDocument* document);
const char* gohub_document_signer_info(GohubDocument* document);
const GohubWChar* gohub_document_signer_info_w(GohubDocument* document);
const char* gohub_document_sign_time(GohubDocument* document);
const GohubWChar* gohub_document_sign_time_w(GohubDocument* document);

////////////////////////////////////
// Накладення електронно-цифрового підпису //
////////////////////////////////////
GohubBool gohub_open_private_key(GohubConnection* connection, const char*
passwordToKey);
GohubBool gohub_open_private_key_w(GohubConnection* connection, const GohubWChar*
passwordToKey);
GohubBool gohub_open_private_key_from_path(GohubConnection* connection, const
char* passwordToKey, const char* keyFileName);
GohubBool gohub_open_private_key_from_path_w(GohubConnection* connection, const
GohubWChar* passwordToKey, const GohubWChar* keyFileName);
GohubBool gohub_open_private_key_by_bytes(GohubConnection* connection, const
char* passwordToKey, unsigned char* keyBinary, unsigned int length);
GohubBool gohub_open_private_key_by_bytes_w(GohubConnection* connection, const
GohubWChar* passwordToKey, unsigned char* keyBinary, unsigned int length);
const char* gohub_private_key_owner_name(GohubConnection* connection);
const GohubWChar* gohub_private_key_owner_name_w(GohubConnection* connection);
const char* gohub_private_key_owner_info(GohubConnection* connection);
const GohubWChar* gohub_private_key_owner_info_w(GohubConnection* connection);
GohubBool gohub_sign_document(GohubConnection* connection, GohubDocument*
document);
GohubBool gohub_sign_fdu92(GohubConnection* connection, GohubFdu92* fdu92);
GohubBool gohub_sign_gu46(GohubConnection* connection, GohubGu46* gu46);
GohubBool gohub_close_private_key(GohubConnection* connection);
GohubBool gohub_delete_old_certs_csk_uz(GohubConnection* connection, const char*
passwordToKey);
GohubBool gohub_delete_old_certs_csk_uz_w(GohubConnection* connection, const
GohubWChar* passwordToKey);

////////////////////////////////////
// Операції з файлами електронних ключів //
////////////////////////////////////
GohubBool gohub_mount_file_key(const char* keyId, const char* path);
GohubBool gohub_mount_file_key_w(const GohubWChar* keyId, const GohubWChar*
path);
GohubBool gohub_unmount_file_key(const char* keyId);
GohubBool gohub_unmount_file_key_w(const GohubWChar* keyId);
int gohub_query_mounted_file_keys();
const char* gohub_mounted_file_key_id(int index);
const GohubWChar* gohub_mounted_file_key_id_w(int index);
const char* gohub_mounted_file_key_dir(int index);
const GohubWChar* gohub_mounted_file_key_dir_w(int index);

////////////////////////////////////
// Обробка помилок //
////////////////////////////////////
GohubErrcode gohub_last_error_code();
const char* gohub_last_error_title();
const GohubWChar* gohub_last_error_title_w();
const char* gohub_last_error_text();
const GohubWChar* gohub_last_error_text_w();

////////////////////////////////////

```



```

// Робота з Гу27 //
////////////////////////////////
GohubGu27* gohub_load_gu27(const char* path);
GohubGu27* gohub_load_gu27_w(const GohubWChar* path);
GohubGu27* gohub_create_gu27(const char* content);
GohubGu27* gohub_create_gu27_w(const GohubWChar* content);
GohubGu27* gohub_query_gu27(GohubConnection* connection, const char* gu27Id);
GohubGu27* gohub_query_gu27_w(GohubConnection* connection, const GohubWChar*
gu27Id);
GohubGu27* gohub_query_next_gu27(GohubConnection* connection, int lastRevision);
const char* gohub_gu27_id(GohubGu27* gu27);
const GohubWChar* gohub_gu27_id_w(GohubGu27* gu27);
int gohub_gu27_revision(GohubGu27* gu27);
const char* gohub_gu27_text(GohubGu27* gu27);
const GohubWChar* gohub_gu27_text_w(GohubGu27* gu27);
const char* gohub_gu27_data_text(GohubGu27* gu27, int gu27Version);
const GohubWChar* gohub_gu27_data_text_w(GohubGu27* gu27, int gu27Version);
int gohub_gu27_size(GohubGu27* gu27);
GohubBool gohub_send_gu27(GohubConnection* connection, GohubGu27* gu27);
GohubBool gohub_send_received_gu27(GohubConnection* connection, GohubGu27* gu27,
const char* gu27Id);
GohubBool gohub_send_received_gu27_w(GohubConnection* connection, GohubGu27*
gu27, const GohubWChar* gu27Id);
GohubBool gohub_save_gu27(GohubGu27* gu27, const char* path, int codePage);
GohubBool gohub_save_gu27_w(GohubGu27* gu27, const GohubWChar* path, int
codePage);
GohubBool gohub_save_gu27_data(GohubGu27* gu27, const char* path, int codePage,
int gu27Version);
GohubBool gohub_save_gu27_data_w(GohubGu27* gu27, const GohubWChar* path, int
codePage, int gu27Version);
GohubBool gohub_close_gu27(GohubGu27* gu27);
GohubBool gohub_reclaim_gu27(GohubConnection* connection, const char* gu27Id);
GohubBool gohub_reclaim_gu27_w(GohubConnection* connection, const GohubWChar*
gu27Id);
GohubBool gohub_delete_gu27(GohubConnection* connection, const char* gu27Id);
GohubBool gohub_delete_gu27_w(GohubConnection* connection, const GohubWChar*
gu27Id);
GohubGu27Status gohub_gu27_status(GohubGu27* gu27);

GohubBool gohub_sign_gu27(GohubConnection* connection, GohubGu27* gu27);

////////////////////////////////
// Перевірка електронно-цифрового підпису GU27 //
////////////////////////////////
GohubBool gohub_gu27_has_signature(GohubGu27* gu27);
GohubBool gohub_gu27_check_signature(GohubGu27* gu27);
const char* gohub_gu27_signer_name(GohubGu27* gu27);
const GohubWChar* gohub_gu27_signer_name_w(GohubGu27* gu27);
const char* gohub_gu27_signer_info(GohubGu27* gu27);
const GohubWChar* gohub_gu27_signer_info_w(GohubGu27* gu27);
const char* gohub_gu27_sign_time(GohubGu27* gu27);
const GohubWChar* gohub_gu27_sign_time_w(GohubGu27* gu27);

////////////////////////////////
// Робота з АС "Месплан" //
////////////////////////////////
const char* gohub_get_mp_months(GohubConnection* connection, int codePage);
const GohubWChar* gohub_get_mp_months_w(GohubConnection* connection);
GohubBool gohub_query_and_save_orders_for_month(GohubConnection* connection,
const char* month, const char* path);
GohubBool gohub_query_and_save_orders_for_month_w(GohubConnection* connection,
const GohubWChar* month, const GohubWChar* path);
GohubBool gohub_query_and_save_orders_for_month_with_relogin(GohubConnection*
connection, const char* month, const char* login, const char* password, const
char* path);

```

```

GohubBool gohub_query_and_save_orders_for_month_with_relogin_w(GohubConnection*
connection, const GohubWChar* month, const GohubWChar* login, const GohubWChar*
password, const GohubWChar* path);

////////////////////////////////////
// Робота з документами інформаційних послуг //
////////////////////////////////////
GohubInformServicesDoc* gohub_query_inform_services_document(GohubConnection*
connection, unsigned __int64 docId);
GohubInformServicesDoc*
gohub_query_next_inform_services_document(GohubConnection* connection, unsigned
__int64 lastRevision);
GohubBool gohub_save_inform_services_document(GohubInformServicesDoc* document,
const char* path);
GohubBool gohub_save_inform_services_document_w(GohubInformServicesDoc* document,
const GohubWChar* path);
GohubBool gohub_saveXml_inform_services_document(GohubInformServicesDoc*
document, const char* path);
GohubBool gohub_saveXml_inform_services_document_w(GohubInformServicesDoc*
document, const GohubWChar* path);
GohubBool gohub_close_inform_services_document(GohubInformServicesDoc* document);
unsigned __int64 gohub_inform_services_document_id(GohubInformServicesDoc*
document);
unsigned __int64 gohub_inform_services_document_revision(GohubInformServicesDoc*
document);
const char* gohub_inform_services_document_filename(GohubInformServicesDoc*
document);
const GohubWChar*
gohub_inform_services_document_filename_w(GohubInformServicesDoc* document);
const char* gohub_inform_services_document_comment(GohubInformServicesDoc*
document);
const GohubWChar*
gohub_inform_services_document_comment_w(GohubInformServicesDoc* document);
const char* gohub_inform_services_document_created_date(GohubInformServicesDoc*
document);
const GohubWChar*
gohub_inform_services_document_created_date_w(GohubInformServicesDoc* document);
const char* gohub_inform_services_document_doc_date(GohubInformServicesDoc*
document);
const GohubWChar*
gohub_inform_services_document_doc_date_w(GohubInformServicesDoc* document);

////////////////////////////////////
// Робота з переліком про замовлення на погодження перевезення за даними
календаря планування перевезень зернових вантажів(за останні 5 днів від поточної
дати) //
////////////////////////////////////
GohubDispatchInfo* gohub_query_dispatch_info(GohubConnection* connection, const
char* start_esr, const char* end_esr);
GohubDispatchInfo* gohub_query_dispatch_info_w(GohubConnection* connection, const
GohubWChar* start_esr, const GohubWChar* end_esr);
const char* gohub_document_info_description(GohubDispatchInfo* document, int
index);
int gohub_document_info_count(GohubDispatchInfo* document);
GohubBool gohub_close_dispatch_info(GohubDispatchInfo* document);
const char* gohub_document_info_number(GohubDispatchInfo* document, int index);
const char* gohub_document_info_type(GohubDispatchInfo* document, int index);
const char* gohub_document_info_date(GohubDispatchInfo* document, int index);
const char* gohub_document_info_wag_owner(GohubDispatchInfo* document, int
index);
const char* gohub_document_info_is_empty(GohubDispatchInfo* document, int
index);

```

```

const GohubWChar* gohub_document_info_number_w(GohubDispatchInfo* document, int
index);
const GohubWChar* gohub_document_info_type_w(GohubDispatchInfo* document, int
index);
const GohubWChar* gohub_document_info_date_w(GohubDispatchInfo* document, int
index);
const GohubWChar* gohub_document_info_wag_owner_w(GohubDispatchInfo* document,
int index);
const GohubWChar* gohub_document_info_is_empty_w(GohubDispatchInfo* document,
int index);
const GohubWChar* gohub_document_info_description_w(GohubDispatchInfo* document,
int index);GohubDispatchInfo* gohub_query_dispatch_info(GohubConnection*
connection, const char* start_esr, const char* end_esr);
GohubDispatchInfo* gohub_query_dispatch_info_w(GohubConnection* connection, const
GohubWChar* start_esr, const GohubWChar* end_esr);
const char* gohub_document_info_description(GohubDispatchInfo* document, int
index);
int gohub_document_info_count(GohubDispatchInfo* document);
GohubBool gohub_close_dispatch_info(GohubDispatchInfo* document);
const char* gohub_document_info_number(GohubDispatchInfo* document, int index);
const char* gohub_document_info_type(GohubDispatchInfo* document, int index);
const char* gohub_document_info_date(GohubDispatchInfo* document, int index);
const char* gohub_document_info_wag_owner(GohubDispatchInfo* document, int
index);
const char* gohub_document_info_is_empty(GohubDispatchInfo* document, int
index);
const GohubWChar* gohub_document_info_number_w(GohubDispatchInfo* document, int
index);
const GohubWChar* gohub_document_info_type_w(GohubDispatchInfo* document, int
index);
const GohubWChar* gohub_document_info_date_w(GohubDispatchInfo* document, int
index);
const GohubWChar* gohub_document_info_wag_owner_w(GohubDispatchInfo* document,
int index);
const GohubWChar* gohub_document_info_is_empty_w(GohubDispatchInfo* document,
int index);
const GohubWChar* gohub_document_info_description_w(GohubDispatchInfo* document,
int index);

#ifdef __cplusplus
} //extern "C"
#endif

#endif //GOHUB_CLIENT_H_

```

## ДОДАТОК Б. Файл *gohub.client.errors.h*

```
#ifndef GOHUB_CLIENT_ERRORS_H_
#define GOHUB_CLIENT_ERRORS_H_

#ifdef __cplusplus
extern "C" {
#endif

//////////
// КОДИ ПОМИЛОК //
//////////

typedef enum GohubErrcode
{
    // Operation is success
    gohub_success,

    // Gohub server connection could not be established
    gohub_server_connection_could_not_be_established,

    // Gohub server inaccessible
    gohub_server_inaccessible,

    // Document creation failed
    gohub_document_creation_failed,

    // Document query failed
    gohub_document_query_failed,

    // Next document query failed
    gohub_next_document_query_failed,

    // Document sending failed
    gohub_document_sending_failed,

    // Document saving failed
    gohub_document_saving_failed,

    // Document loading failed
    gohub_document_loading_failed,

    // Invalid code page
    gohub_invalid_code_page,

    // Private key could not be opened
    gohub_private_key_could_not_be_opened,

    // Private key path could not be opened
    gohub_private_key_path_could_not_be_opened,

    // Private key bytes could not be opened
    gohub_private_key_bytes_could_not_be_opened,

    // Private key is inaccessible
    gohub_private_key_is_inaccessible,

    // Document could not be signed
    gohub_document_could_not_be_signed,

    // Document signature verification failed
    gohub_document_signature_verification_failed,

    // Document has not signature
    gohub_document_has_not_signature,
```

```
// Gohub Client is obsolete. Upgrade your Gohub Client version
gohub_client_is_obsolete,

// Gohub Server is obsolete. Upgrade your Gohub Server version
gohub_server_is_obsolete,

// Document reclamation failed
gohub_document_reclamation_failed,

// Document deletion failed
gohub_document_deletion_failed,

// Attachment creation failed
gohub_attachment_creation_failed,

// Attachment sending failed
gohub_attachment_sending_failed,

// Attachment query failed
gohub_attachment_query_failed,

// Attachment deletion failed
gohub_attachment_deletion_failed,

// Mount of file key failed
gohub_mount_of_file_key_failed,

// Unmount of file key failed
gohub_unmount_of_file_key_failed,

// Enumerating of file keys failed
gohub_enumerating_of_file_keys_failed,

// Mounted file key inaccessible
gohub_mounted_file_key_inaccessible,

// EData creation failed
gohub_edata_creation_failed,

// EData sending failed
gohub_edata_sending_failed,

// EData updating failed
gohub_edata_updating_failed,

// EData query failed
gohub_edata_query_failed,

// Next EData query failed
gohub_next_edata_query_failed,

// PiPackage query failed
gohub_pipackage_query_failed,

// Next PiPackage query failed
gohub_next_pipackage_query_failed,

// MP months query failed
gohub_mp_months_query_failed,

// Orders of month query failed
gohub_orders_of_months_query_failed,

// Internal program error
```

```
gohub_programm_error = 0x1000,

// Client application performed invalid operation
gohub_invalid_operation,

// "Inform services doc saving failed"
gohub_inform_services_doc_saving_failed,

// "Inform services doc query failed"
gohub_inform_services_doc_query_failed,

// "Query changes inform services failed"]
gohub_query_changes_inform_services_failed,

// "Query next inform services document"]
gohub_query_next_inform_services_document_failed,

} GohubErrcode;

#ifdef __cplusplus
} //extern "C"
#endif

#endif //GOHUB_CLIENT_ERRORS_H_
```

## ДОДАТОК В. Файл опису СОМ-інтерфейсів GohubClientCOM.idl

```
import "oaidl.idl";
import "ocidl.idl";

interface IGohubClient;
interface IGohubDocument;
interface IGohubAttachment;
interface IGohubEData;
interface IGohubPiPackage;
interface IGohubPiPackageToEData;
interface IGohubConnection;
interface IGohubSignerInfo;
interface IGohubError;
interface IGohubFdu92;
interface IGohubGu46;
interface IGohubGu45;
interface IGohubGu27;
interface IGohubInformServicesDocument;

[
    object,
    uuid(B0E1F579-5836-4E97-9340-58B092418947),
    dual,
    nonextensible,
    helpstring("IGohubSignerInfo Interface"),
    pointer_default(unique)
]
interface IGohubSignerInfo : IDispatch{
    [propget, id(1), helpstring("property Name")] HRESULT Name([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property Subject")] HRESULT Subject([out, retval]
BSTR* pVal);
};
[
    object,
    uuid(0D5D6225-8E07-46E2-8B8D-9C0966481994),
    dual,
    nonextensible,
    helpstring("IGohubDocument Interface"),
    pointer_default(unique)
]
interface IGohubDocument : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property Text")] HRESULT Text([out, retval] BSTR*
pVal);
    [propget, id(3), helpstring("property Revision")] HRESULT Revision([out,
retval] LONG* pVal);
    [propget, id(4), helpstring("property HasSignature")] HRESULT
HasSignature([out, retval] VARIANT_BOOL* pVal);
    [propget, id(5), helpstring("property Signer")] HRESULT Signer([out, retval]
IGohubSignerInfo** pVal);
    [propget, id(6), helpstring("property SignTime")] HRESULT SignTime([out,
retval] BSTR* pVal);
    [id(7), helpstring("method Save")] HRESULT Save([in] BSTR path, [in] LONG
codePage, [out,retval] VARIANT_BOOL* result);
    [id(8), helpstring("method Sign")] HRESULT Sign([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(9), helpstring("method Send")] HRESULT Send([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(10), helpstring("method Close")] HRESULT Close(void);
    [propget, id(11), helpstring("property AttachmentsCount")] HRESULT
AttachmentsCount([out, retval] LONG* pVal);
}
```

```

[id(12), helpstring("method GetAttachmentIdByIndex")] HRESULT
GetAttachmentIdByIndex([in] LONG index, [out,retval] BSTR* result);
[propget, id(15), helpstring("property Status")] HRESULT Status([out, retval]
LONG* pVal);
[id(16), helpstring("method SendReceived")] HRESULT SendReceived([in]
IGohubConnection* connection, [in] BSTR docId, [out,retval] VARIANT_BOOL*
result);
[id(17), helpstring("method SaveData")] HRESULT SaveData([in] BSTR path, [in]
LONG codePage, [in] LONG epdVersion, [out,retval] VARIANT_BOOL* result);
[id(18), helpstring("method DataText")] HRESULT DataText([in] LONG epdVersion,
[out, retval] BSTR* pVal);
[propget, id(19), helpstring("property MeasureEquipNum")] HRESULT
MeasureEquipNum([out, retval] BSTR* pVal);
[propget, id(19), helpstring("property MeasureEquipNum")] HRESULT
MeasureEquipNum([in] BSTR pVal);
[id(20), helpstring("method set_VerifiedEmptyWeightForWagon")] HRESULT
set_VerifiedEmptyWeightForWagon([in] LONG wagonIndex, [in] LONG weight);
[id(21), helpstring("method get_VerifiedEmptyWeightForWagon")] HRESULT
get_VerifiedEmptyWeightForWagon([in] LONG wagonIndex, [out,retval] LONG* weight);
[propget, id(22), helpstring("property BusinessUnitNum")] HRESULT
BusinessUnitNum([out, retval] BSTR* pVal);
[propget, id(22), helpstring("property BusinessUnitNum")] HRESULT
BusinessUnitNum([in] BSTR pVal);
[propget, id(23), helpstring("property ForeignNotAccept")] HRESULT
ForeignNotAccept([out, retval] VARIANT_BOOL* pVal);
[propget, id(24), helpstring("property WarrantType")] HRESULT WarrantType([out,
retval] LONG* pVal);
[propget, id(24), helpstring("property WarrantType")] HRESULT WarrantType([in]
LONG newVal);
[id(25), helpstring("method SetVerifiedEmptyWeightForWagon")] HRESULT
SetVerifiedEmptyWeightForWagon([in] LONG wagonIndex, [in] LONG weight);
[id(26), helpstring("method GetVerifiedEmptyWeightForWagon")] HRESULT
GetVerifiedEmptyWeightForWagon([in] LONG wagonIndex, [out,retval] LONG* weight);
[id(27), helpstring("method GetOTPR")] HRESULT GetOTPR([in] BSTR path,
[out,retval] VARIANT_BOOL* result);
[propget, id(28), helpstring("property OTPRString")] HRESULT OTPRString([out,
retval] BSTR* pVal);
[propget, id(29), helpstring("property Warning")] HRESULT Warning([out, retval]
BSTR* pVal);
};
[
    object,
    uuid(E9967B9D-1141-47BA-A5C4-573FB02DB396),
    dual,
    nonextensible,
    helpstring("IGohubConnection Interface"),
    pointer_default(unique)
]
interface IGohubConnection : IDispatch{
    [propget, id(1), helpstring("property Signer")] HRESULT Signer([out, retval]
IGohubSignerInfo** pVal);
    [id(2), helpstring("method Close")] HRESULT Close(void);
    [id(3), helpstring("method QueryDocument")] HRESULT QueryDocument([in] BSTR
docId, [out,retval] IGohubDocument** document);
    [id(4), helpstring("method QueryNextDocument")] HRESULT QueryNextDocument([in]
LONG revision, [out,retval] IGohubDocument** document);
    [id(5), helpstring("method OpenPrivateKey")] HRESULT OpenPrivateKey([in] BSTR
password, [out,retval] VARIANT_BOOL* result);
    [id(6), helpstring("method ReclaimDocument")] HRESULT ReclaimDocument([in] BSTR
docId, [out,retval] VARIANT_BOOL* result);
    [id(7), helpstring("method DeleteDocument")] HRESULT DeleteDocument([in] BSTR
docId, [out,retval] VARIANT_BOOL* result);
    [id(8), helpstring("method QueryAttachment")] HRESULT QueryAttachment([in] BSTR
attachmentId, [out,retval] IGohubAttachment** attachment);

```



```

[id(9), helpstring("method DeleteAttachment")] HRESULT DeleteAttachment([in]
BSTR attachmentId, [out,retval] VARIANT_BOOL* result);
[propget, id(10), helpstring("property FilterByDocumentStatus")] HRESULT
FilterByDocumentStatus([out, retval] LONG* pVal);
[propput, id(10), helpstring("property FilterByDocumentStatus")] HRESULT
FilterByDocumentStatus([in] LONG newVal);
[propget, id(11), helpstring("property FilterByDocumentNumber")] HRESULT
FilterByDocumentNumber([out, retval] BSTR* pVal);
[propput, id(11), helpstring("property FilterByDocumentNumber")] HRESULT
FilterByDocumentNumber([in] BSTR newVal);
[propget, id(12), helpstring("property FilterByWagonNumber")] HRESULT
FilterByWagonNumber([out, retval] BSTR* pVal);
[propput, id(12), helpstring("property FilterByWagonNumber")] HRESULT
FilterByWagonNumber([in] BSTR newVal);
[propget, id(13), helpstring("property FilterByDepartureClientCode")] HRESULT
FilterByDepartureClientCode([out, retval] BSTR* pVal);
[propput, id(13), helpstring("property FilterByDepartureClientCode")] HRESULT
FilterByDepartureClientCode([in] BSTR newVal);
[propget, id(14), helpstring("property FilterByDeparturePayerCode")] HRESULT
FilterByDeparturePayerCode([out, retval] BSTR* pVal);
[propput, id(14), helpstring("property FilterByDeparturePayerCode")] HRESULT
FilterByDeparturePayerCode([in] BSTR newVal);
[propget, id(15), helpstring("property FilterByDepartureStationCode")] HRESULT
FilterByDepartureStationCode([out, retval] BSTR* pVal);
[propput, id(15), helpstring("property FilterByDepartureStationCode")] HRESULT
FilterByDepartureStationCode([in] BSTR newVal);
[propget, id(16), helpstring("property FilterByArrivalClientCode")] HRESULT
FilterByArrivalClientCode([out, retval] BSTR* pVal);
[propput, id(16), helpstring("property FilterByArrivalClientCode")] HRESULT
FilterByArrivalClientCode([in] BSTR newVal);
[propget, id(17), helpstring("property FilterByArrivalPayerCode")] HRESULT
FilterByArrivalPayerCode([out, retval] BSTR* pVal);
[propput, id(17), helpstring("property FilterByArrivalPayerCode")] HRESULT
FilterByArrivalPayerCode([in] BSTR newVal);
[propget, id(18), helpstring("property FilterByArrivalStationCode")] HRESULT
FilterByArrivalStationCode([out, retval] BSTR* pVal);
[propput, id(18), helpstring("property FilterByArrivalStationCode")] HRESULT
FilterByArrivalStationCode([in] BSTR newVal);
[id(19), helpstring("method ClearAllFilters")] HRESULT
ClearAllFilters([out,retval] VARIANT_BOOL* result);
[id(20), helpstring("method QueryAndSaveDocumentPrintableForm")] HRESULT
QueryAndSaveDocumentPrintableForm([in] BSTR docId, [in] BSTR path, [out,retval]
VARIANT_BOOL* result);
[id(21), helpstring("method QueryAndSaveFdu92PrintableForm")] HRESULT
QueryAndSaveFdu92PrintableForm([in] BSTR docId, [in] BSTR path, [out,retval]
VARIANT_BOOL* result);
[id(22), helpstring("method QueryAndSaveGu45PrintableForm")] HRESULT
QueryAndSaveGu45PrintableForm([in] BSTR docId, [in] BSTR path, [out,retval]
VARIANT_BOOL* result);
[id(23), helpstring("method QueryAndSaveGu46PrintableForm")] HRESULT
QueryAndSaveGu46PrintableForm([in] BSTR docId, [in] BSTR path, [out,retval]
VARIANT_BOOL* result);
[id(24), helpstring("method QueryFdu92")] HRESULT QueryFdu92([in] BSTR id,
[out,retval] IGohubFdu92** fdu92);
[id(25), helpstring("method QueryNextFdu92")] HRESULT QueryNextFdu92([in] LONG
revision, [out,retval] IGohubFdu92** fdu92);
[id(26), helpstring("method QueryGu46")] HRESULT QueryGu46([in] BSTR id,
[out,retval] IGohubGu46** gu46);
[id(27), helpstring("method QueryNextGu46")] HRESULT QueryNextGu46([in] LONG
revision, [out,retval] IGohubGu46** gu46);
[id(28), helpstring("method QueryGu45")] HRESULT QueryGu45([in] BSTR id,
[out,retval] IGohubGu45** gu45);
[id(29), helpstring("method QueryNextGu45")] HRESULT QueryNextGu45([in] LONG
revision, [out,retval] IGohubGu45** gu45);
[id(30), helpstring("method RejectFdu92")] HRESULT RejectFdu92(BSTR id);

```

```

    [id(31), helpstring("method RejectGu46")] HRESULT RejectGu46(BSTR id);
    [id(32), helpstring("method QueryEData")] HRESULT QueryEData([in] BSTR eDataId,
[out,retval] IGohubEData** eData);
    [id(33), helpstring("method QueryNextEData")] HRESULT QueryNextEData([in]
ULONGLONG revision, [out,retval] IGohubEData** eData);
    [id(34), helpstring("method QueryPiPackage")] HRESULT QueryPiPackage([in] BSTR
eDataId, [out,retval] IGohubPiPackage** eData);
    [id(35), helpstring("method QueryNextPiPackage")] HRESULT
QueryNextPiPackage([in] ULONGLONG revision, [out,retval] IGohubPiPackage**
eData);
    [id(36), helpstring("method AddEDDataToPiPackage")] HRESULT
AddEDDataToPiPackage([in] IGohubEData* eData, [in] BSTR piPackageId, [out,retval]
IGohubPiPackage** piPackage);
    [id(37), helpstring("method QueryGu27")] HRESULT QueryGu27([in] BSTR gu27Id,
[out,retval] IGohubGu27** gu27);
    [id(38), helpstring("method QueryNextGu27")] HRESULT QueryNextGu27([in] LONG
revision, [out,retval] IGohubGu27** gu27);
    [id(39), helpstring("method ReclaimGu27")] HRESULT ReclaimGu27([in] BSTR
gu27Id, [out,retval] VARIANT_BOOL* result);
    [id(40), helpstring("method DeleteGu27")] HRESULT DeleteGu27([in] BSTR gu27Id,
[out,retval] VARIANT_BOOL* result);
    [id(41), helpstring("method QueryAndSaveGu27PrintableForm")] HRESULT
QueryAndSaveGu27PrintableForm([in] BSTR gu27Id, [in] BSTR path, [out,retval]
VARIANT_BOOL* result);
    [id(42), helpstring("method QueryAttachmentWithUserData")] HRESULT
QueryAttachmentWithUserData([in] BSTR attachmentId, [out,retval]
IGohubAttachment** attachment);
    [id(43), helpstring("method GetMPMonths")] HRESULT GetMPMonths([out,retval]
BSTR* result);
    [id(44), helpstring("method QueryAndSaveOrdersForMonth")] HRESULT
QueryAndSaveOrdersForMonth([in] BSTR month, [in] BSTR path);
    [id(45), helpstring("method QueryAndSaveOrdersForMonthWithReLogin")] HRESULT
QueryAndSaveOrdersForMonthWithReLogin([in] BSTR month, [in] BSTR login, [in] BSTR
password, [in] BSTR path);
    [id(46), helpstring("method SaveDocumentPrintableForm")] HRESULT
SaveDocumentPrintableForm([in] BSTR docId, [in] BSTR path, [out,retval]
VARIANT_BOOL* result);
    [id(47), helpstring("method SaveOrdersForMonthWithReLogin")] HRESULT
SaveOrdersForMonthWithReLogin([in] BSTR month, [in] BSTR login, [in] BSTR
password, [in] BSTR path);
    [id(48), helpstring("method QueryEDDataForAttachment")] HRESULT
QueryEDDataForAttachment([in] BSTR attachmentId, [out,retval] IGohubEData**
eData);
    [id(49), helpstring("method QueryInfServsDoc")] HRESULT QueryInfServsDoc([in]
ULONGLONG docId, [out,retval] IGohubInformServicesDocument** document);
    [id(50), helpstring("method QueryNextInfServsDoc")] HRESULT
QueryNextInfServsDoc([in] ULONGLONG revision, [out,retval]
IGohubInformServicesDocument** document);
    [id(51), helpstring("method OpenPrivateKeyFromPath")] HRESULT
OpenPrivateKeyFromPath([in] BSTR password, [in] BSTR path, [out,retval]
VARIANT_BOOL* result);
    [id(52), helpstring("method QueryFdu92ByNumber")] HRESULT
QueryFdu92ByNumber([in] BSTR registration_esr, [in] BSTR registration_num,
[out,retval] IGohubFdu92** fdu92);
    [id(53), helpstring("method QueryDispatchInfo")] HRESULT QueryDispatchInfo([in]
BSTR dispatch_esr, [in] BSTR arrival_num, [out,retval] IGohubDispatchInfo** disp);
};

[
    object,
    uuid(F1077417-21D9-4871-84A2-9F89525E7214),
    dual,
    nonextensible,
    helpstring("IGohubError Interface"),
    pointer_default(unique)
]

```

```

interface IGohubError : IDispatch{
    [propget, id(1), helpstring("property Code")] HRESULT Code([out, retval] LONG*
pVal);
    [propget, id(2), helpstring("property Title")] HRESULT Title([out, retval]
BSTR* pVal);
    [propget, id(3), helpstring("property Text")] HRESULT Text([out, retval] BSTR*
pVal);
};
[
    object,
    uuid(ABDA6C07-5320-4F28-B995-FADE037D0A82),
    dual,
    nonextensible,
    helpstring("IGohubClient Interface"),
    pointer_default(unique)
]
interface IGohubClient : IDispatch{
    [id(1), helpstring("method GetLastError")] HRESULT GetLastError([out,retval]
IGohubError** result);
    [id(2), helpstring("method Connect")] HRESULT Connect([in] BSTR host, [in] LONG
port, [out,retval] IGohubConnection** connection);
    [id(3), helpstring("method CreateDocument")] HRESULT CreateDocument([in] BSTR
xmlText, [out,retval] IGohubDocument** document);
    [id(4), helpstring("method LoadDocument")] HRESULT LoadDocument([in] BSTR path,
[out,retval] IGohubDocument** result);
    [id(5), helpstring("method LoadAttachment")] HRESULT LoadAttachment([in] BSTR
typeCode, [in] BSTR name, [in] BSTR regNumber, [in] BSTR regDate, [in] BSTR
validFrom, [in] BSTR validTo, [in] BSTR path, [out,retval] IGohubAttachment**
result);
    [id(6), helpstring("method MountFileKey")] HRESULT MountFileKey([in] BSTR
keyId, [in] BSTR keyDir, [out,retval] VARIANT_BOOL* result);
    [id(7), helpstring("method UnmountFileKey")] HRESULT UnmountFileKey([in] BSTR
keyId, [out,retval] VARIANT_BOOL* result);
    [id(8), helpstring("method QueryMountedKeys")] HRESULT
QueryMountedKeys([out,retval] LONG* result);
    [id(9), helpstring("method GetMountedKeyId")] HRESULT GetMountedKeyId([in] LONG
index, [out,retval] BSTR* keyId);
    [id(10), helpstring("method GetMountedKeyDir")] HRESULT GetMountedKeyDir([in]
LONG index, [out,retval] BSTR* keyDir);
    [id(11), helpstring("method LoadEData")] HRESULT LoadEData([in] UINT codeType,
[in] BSTR xmlPath, [in] BSTR name, [in] BSTR regNumber, [in] BSTR regDate, [in]
BSTR validFrom, [in] BSTR validTo, [in] BSTR pdfPath, [out,retval] IGohubEData**
result);
    [id(12), helpstring("method LoadEDataSimple")] HRESULT LoadEDataSimple([in]
UINT codeType, [in] BSTR xmlPath, [out,retval] IGohubEData** result);
    [id(13), helpstring("method CreateGu27")] HRESULT CreateGu27([in] BSTR xmlText,
[out,retval] IGohubGu27** gu27);
    [id(14), helpstring("method LoadGu27")] HRESULT LoadGu27([in] BSTR path,
[out,retval] IGohubGu27** result);
    [id(15), helpstring("method LoadSmgsAttachment")] HRESULT
LoadSmgsAttachment([in] BSTR smgsTypeCode, [in] BSTR name, [in] BSTR regNumber,
[in] BSTR regDate, [in] BSTR validFrom, [in] BSTR validTo, [in] BSTR path,
[out,retval] IGohubAttachment** result);
    [id(16), helpstring("method LoadAttachmentWithUserData")] HRESULT
LoadAttachmentWithUserData([in] BSTR typeCode, [in] BSTR name, [in] BSTR
regNumber, [in] BSTR regDate, [in] BSTR validFrom, [in] BSTR validTo, [in] BSTR
path, [in] BSTR pathUserData, [out,retval] IGohubAttachment** attachment);
    [id(17), helpstring("method LoadSmgsAttachmentWithUserData")] HRESULT
LoadSmgsAttachmentWithUserData([in] BSTR smgsTypeCode, [in] BSTR name, [in] BSTR
regNumber, [in] BSTR regDate, [in] BSTR validFrom, [in] BSTR validTo, [in] BSTR
path, [in] BSTR pathUserData, [out,retval] IGohubAttachment** attachment);
    [id(18), helpstring("method LoadFdu92")] HRESULT LoadFdu92([in] BSTR ident,
[in] BSTR path, [out,retval] IGohubFdu92** result);
    [id(19), helpstring("method LoadGu46")] HRESULT LoadGu46([in] BSTR ident, [in]
BSTR path, [out,retval] IGohubGu46** result);

```

```

};
[
    object,
    uuid(044603D2-574E-463C-87EC-DCD98C30F319),
    dual,
    nonextensible,
    helpstring("IGohubAttachment Interface"),
    pointer_default(unique)
]
interface IGohubAttachment : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property TypeCode")] HRESULT TypeCode([out,
retval] BSTR* pVal);
    [propget, id(3), helpstring("property Name")] HRESULT Name([out, retval] BSTR*
pVal);
    [propget, id(4), helpstring("property Description")] HRESULT Description([out,
retval] BSTR* pVal);
    [propget, id(5), helpstring("property RegNumber")] HRESULT RegNumber([out,
retval] BSTR* pVal);
    [propget, id(6), helpstring("property RegDate")] HRESULT RegDate([out, retval]
BSTR* pVal);
    [propget, id(7), helpstring("property ValidFrom")] HRESULT ValidFrom([out,
retval] BSTR* pVal);
    [propget, id(8), helpstring("property ValidTo")] HRESULT ValidTo([out, retval]
BSTR* pVal);
    [id(9), helpstring("method Save")] HRESULT Save([in] BSTR path, [out, retval]
VARIANT_BOOL* result);
    [id(10), helpstring("method Send")] HRESULT Send([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(11), helpstring("method Close")] HRESULT Close(void);
    [propget, id(12), helpstring("property SmgsTypeCode")] HRESULT
SmgsTypeCode([out, retval] BSTR* pVal);
    [id(13), helpstring("method SaveUserData")] HRESULT SaveUserData([in] BSTR
path, [out,retval] VARIANT_BOOL* result);
};
[
    object,
    uuid(685B21FA-43A7-4ACC-9A57-AB7AC332942C),
    dual,
    nonextensible,
    helpstring("IGohubEData Interface"),
    pointer_default(unique)
]
interface IGohubEData : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property Revision")] HRESULT Revision([out,
retval] ULONGLONG* pVal);
    [propget, id(3), helpstring("property RevisionDate")] HRESULT
RevisionDate([out, retval] BSTR* pVal);
    [propget, id(4), helpstring("property Version")] HRESULT Version([out, retval]
BSTR* pVal);
    [propget, id(5), helpstring("property DocType")] HRESULT DocType([out, retval]
UINT* pVal);
    [propget, id(6), helpstring("property Status")] HRESULT Status([out, retval]
INT* pVal);
    [propget, id(7), helpstring("property AttachmentId")] HRESULT
AttachmentId([out, retval] BSTR* pVal);
    [id(8), helpstring("method Save")] HRESULT Save([in] BSTR path, [out, retval]
VARIANT_BOOL* result);
    [id(9), helpstring("method Send")] HRESULT Send([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(10), helpstring("method LoadData")] HRESULT LoadData([in] BSTR path, [out,
retval] VARIANT_BOOL* result);

```

```

    [id(11), helpstring("method Update")] HRESULT Update([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(12), helpstring("method Close")] HRESULT Close(void);
};
[
    object,
    uuid(62C21013-07D6-4d9d-83C4-9FA6E770B2EA),
    dual,
    nonextensible,
    helpstring("IGohubPiPackage Interface"),
    pointer_default(unique)
]
interface IGohubPiPackage : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property Revision")] HRESULT Revision([out,
retval] ULONGLONG* pVal);
    [propget, id(3), helpstring("property RevisionDate")] HRESULT
RevisionDate([out, retval] BSTR* pVal);
    [propget, id(4), helpstring("property ConsignmentId")] HRESULT
ConsignmentId([out, retval] BSTR* pVal);
    [propget, id(5), helpstring("property Status")] HRESULT Status([out, retval]
INT* pVal);
    [propget, id(6), helpstring("property PiPackageToEDataCount")] HRESULT
PiPackageToEDataCount([out, retval] INT* pVal);
    [id(7), helpstring("method PiPackageToEData")] HRESULT PiPackageToEData([in]
INT index, [out, retval] IGohubPiPackageToEData** result);
    [id(8), helpstring("method Save")] HRESULT Save([in] BSTR path, [out, retval]
VARIANT_BOOL* result);
    [id(10), helpstring("method Close")] HRESULT Close(void);
};
[
    object,
    uuid(88AAFC89-0426-4551-BD1C-F57F63D0C335),
    dual,
    nonextensible,
    helpstring("IGohubPiPackageToEData Interface"),
    pointer_default(unique)
]
interface IGohubPiPackageToEData : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property EDataId")] HRESULT EDataId([out, retval]
BSTR* pVal);
    [propget, id(3), helpstring("property PiPackageId")] HRESULT PiPackageId([out,
retval] BSTR* pVal);
    [propget, id(4), helpstring("property Note")] HRESULT Note([out, retval] BSTR*
pVal);
    [propget, id(5), helpstring("property Status")] HRESULT Status([out, retval]
INT* pVal);
    [propget, id(6), helpstring("property EDataVersion")] HRESULT
EDataVersion([out, retval] BSTR* pVal);
};
[
    object,
    uuid(EB03BE7D-48B7-4AFD-8A94-A5012D844A17),
    dual,
    nonextensible,
    helpstring("IGohubFdu92 Interface"),
    pointer_default(unique)
]
interface IGohubFdu92 : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);

```

```

        [propget, id(2), helpstring("property Status")] HRESULT Status([out,
retval] LONG* pVal);
        [propget, id(3), helpstring("property Revision")] HRESULT Revision([out,
retval] LONG* pVal);
        [propget, id(4), helpstring("property Text")] HRESULT Text([out, retval]
BSTR* pVal);
        [id(5), helpstring("method Save")] HRESULT Save([in] BSTR path, [in] LONG
codePage, [out,retval] VARIANT_BOOL* result);
        [id(6), helpstring("method Close")] HRESULT Close(void);
        [id(7), helpstring("method Sign")] HRESULT Sign([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
        [id(8), helpstring("method Send")] HRESULT Send([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
        [propget, id(9), helpstring("property HasSignature")] HRESULT
HasSignature([out, retval] VARIANT_BOOL* result);
        [propget, id(10), helpstring("property Signer")] HRESULT Signer([out,
retval] IGohubSignerInfo** pVal);
        [propget, id(11), helpstring("property SignTime")] HRESULT SignTime([out,
retval] BSTR* pVal);
    };
    [
        object,
        uuid(78DCD121-84B7-4D41-B15C-F9F7677A3519),
        dual,
        nonextensible,
        helpstring("IGohubGu46 Interface"),
        pointer_default(unique)
    ]
interface IGohubGu46 : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property Status")] HRESULT Status([out,
retval] LONG* pVal);
    [propget, id(3), helpstring("property Revision")] HRESULT Revision([out,
retval] LONG* pVal);
    [propget, id(4), helpstring("property Text")] HRESULT Text([out, retval]
BSTR* pVal);
    [id(5), helpstring("method Save")] HRESULT Save([in] BSTR path, [in] LONG
codePage, [out,retval] VARIANT_BOOL* result);
    [id(6), helpstring("method Close")] HRESULT Close(void);
    [id(7), helpstring("method Sign")] HRESULT Sign([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(8), helpstring("method Send")] HRESULT Send([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [propget, id(9), helpstring("property HasSignature")] HRESULT
HasSignature([out, retval] VARIANT_BOOL* result);
    [propget, id(10), helpstring("property Signer")] HRESULT Signer([out,
retval] IGohubSignerInfo** pVal);
    [propget, id(11), helpstring("property SignTime")] HRESULT SignTime([out,
retval] BSTR* pVal);
};
    [
        object,
        uuid(C2F92D3C-295A-4453-B4CF-B33D2C8966E3),
        dual,
        nonextensible,
        helpstring("IGohubGu45 Interface"),
        pointer_default(unique)
    ]
interface IGohubGu45 : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property Status")] HRESULT Status([out,
retval] LONG* pVal);

```



```

        [propget, id(3), helpstring("property Revision")] HRESULT Revision([out,
retval] LONG* pVal);
        [propget, id(4), helpstring("property Text")] HRESULT Text([out, retval]
BSTR* pVal);
        [id(5), helpstring("method Save")] HRESULT Save([in] BSTR path, [in] LONG
codePage, [out,retval] VARIANT_BOOL* result);
        [id(6), helpstring("method Close")] HRESULT Close(void);
        [propget, id(7), helpstring("property HasSignature")] HRESULT
HasSignature([out, retval] VARIANT_BOOL* result);
        [propget, id(8), helpstring("property Signer")] HRESULT Signer([out,
retval] IGohubSignerInfo** pVal);
        [propget, id(9), helpstring("property SignTime")] HRESULT SignTime([out,
retval] BSTR* pVal);
    };
    [
        object,
        uuid(4B840AC2-7128-4989-A56C-2570E8435C48),
        dual,
        nonextensible,
        helpstring("IGohubGu27 Interface"),
        pointer_default(unique)
    ]
interface IGohubGu27 : IDispatch{
    [propget, id(1), helpstring("property Id")] HRESULT Id([out, retval] BSTR*
pVal);
    [propget, id(2), helpstring("property Text")] HRESULT Text([out, retval]
BSTR* pVal);
    [propget, id(3), helpstring("property Revision")] HRESULT Revision([out,
retval] LONG* pVal);
    [propget, id(4), helpstring("property HasSignature")] HRESULT
HasSignature([out, retval] VARIANT_BOOL* pVal);
    [propget, id(5), helpstring("property Signer")] HRESULT Signer([out,
retval] IGohubSignerInfo** pVal);
    [propget, id(6), helpstring("property SignTime")] HRESULT SignTime([out,
retval] BSTR* pVal);
    [id(7), helpstring("method Save")] HRESULT Save([in] BSTR path, [in] LONG
codePage, [out,retval] VARIANT_BOOL* result);
    [id(8), helpstring("method Sign")] HRESULT Sign([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(9), helpstring("method Send")] HRESULT Send([in] IGohubConnection*
connection, [out,retval] VARIANT_BOOL* result);
    [id(10), helpstring("method Close")] HRESULT Close(void);
    [propget, id(11), helpstring("property Status")] HRESULT Status([out,
retval] LONG* pVal);
    [id(12), helpstring("method SendReceived")] HRESULT SendReceived([in]
IGohubConnection* connection, [in] BSTR gu27Id, [out,retval] VARIANT_BOOL*
result);
    [id(13), helpstring("method SaveData")] HRESULT SaveData([in] BSTR path,
[in] LONG codePage, [in] LONG gu27Version, [out,retval] VARIANT_BOOL* result);
    [id(14), helpstring("method DataText")] HRESULT DataText([in] LONG
gu27Version, [out,retval] BSTR* pVal);
};
    [
        object,
        uuid(0af49642-b12d-4fdb-b363-a7497cc78462),
        dual,
        nonextensible,
        helpstring("IInformServicesDocument Interface"),
        pointer_default(unique)
    ]
interface IGohubInformServicesDocument : IDispatch{
    [id(1), helpstring("method Save")] HRESULT Save([in] BSTR path,
[out,retval] VARIANT_BOOL* result);
    [id(2), helpstring("method Close")] HRESULT Close(void);

```

```

        [propget, id(3), helpstring("property Id")] HRESULT Id([out, retval]
ULONGLONG* pVal);
        [propget, id(4), helpstring("property Revision")] HRESULT Revision([out,
retval] ULONGLONG* pVal);
        [propget, id(5), helpstring("property FileName")] HRESULT FileName([out,
retval] BSTR* pVal);
        [propget, id(6), helpstring("property Comment")] HRESULT Comment([out,
retval] BSTR* pVal);
        [propget, id(7), helpstring("property CreatedDate")] HRESULT
CreatedDate([out, retval] BSTR* pVal);
        [propget, id(8), helpstring("property DocDate")] HRESULT DocDate([out,
retval] BSTR* pVal);
        [id(9), helpstring("method SaveXml")] HRESULT SaveXml([in] BSTR path,
[out,retval] VARIANT_BOOL* result);
};
[
    object,
    uuid(0669AB3B-8E1B-4161-8A2E-244D5A62029A),
    dual,
    nonextensible,
    helpstring("IDispatchInfo Interface"),
    pointer_default(unique)
]
interface IGohubDispatchInfo : IDispatch{
    [id(1), helpstring("method Close")] HRESULT Close(void);
    [id(2), helpstring("method WagOwnerByIndex")] HRESULT WagOwnerByIndex([in]
ULONGLONG index, [out,retval] BSTR* pVal);
    [id(3), helpstring("method TypeByIndex")] HRESULT TypeByIndex([in]
ULONGLONG index, [out,retval] BSTR* pVal);
    [id(4), helpstring("method DateByIndex")] HRESULT DateByIndex([in]
ULONGLONG index, [out,retval] BSTR* pVal);
    [id(5), helpstring("method DescriptionByIndex")] HRESULT
DescriptionByIndex([in] ULONGLONG index, [out,retval] BSTR* pVal);
    [id(6), helpstring("method NumberByIndex")] HRESULT NumberByIndex([in]
ULONGLONG index, [out,retval] BSTR* pVal);
    [id(7), helpstring("method IsEmptyByIndex")] HRESULT IsEmptyByIndex([in]
ULONGLONG index, [out,retval] BSTR* pVal);
    [id(8), helpstring("method Count")] HRESULT Count([out,retval] LONG*
count);
};
[
    uuid(EB746ACA-C81E-42EE-B1C4-C435A8CD0082),
    version(1.0),
    helpstring("gohubclientcom 1.0 Type Library")
]
library gohubclientcomLib
{
    importlib("stdole2.tlb");
    [
        uuid(B1166D0A-F0FB-4526-803C-1D37F8EB5100),
        helpstring("GohubDocument Class")
    ]
    coclass GohubComDocument
    {
        [default] interface IGohubDocument;
    };
    [
        uuid(77E0C042-770C-4EA1-9593-1E370E5E8EE3),
        helpstring("GohubConnection Class")
    ]
    coclass GohubComConnection
    {
        [default] interface IGohubConnection;
    };
    [

```



```

        uuid(CF908D67-DAF6-43B0-9621-1DD417CFF3D7),
        helpstring("GohubClient Class")
    ]
coclass GohubComClient
{
    [default] interface IGohubClient;
};
[
    uuid(D3160E70-00DC-4502-A84B-840D6929D99A),
    helpstring("GohubError Class")
]
coclass GohubComError
{
    [default] interface IGohubError;
};
[
    uuid(A86B2DEA-CFC4-48F5-B904-73D638E73F95),
    helpstring("GohubSignerInfo Class")
]
coclass GohubComSignerInfo
{
    [default] interface IGohubSignerInfo;
};
[
    uuid(C1B3AEDC-6673-470D-AFF0-48A0659A9BCD),
    helpstring("GohubAttachment Class")
]
coclass GohubComAttachment
{
    [default] interface IGohubAttachment;
};
[
    uuid(934AD530-23EA-481E-A365-CD37EC536474),
    helpstring("GohubComEData Class")
]
coclass GohubComEData
{
    [default] interface IGohubEData;
};
[
    uuid(72819B78-CC37-4bc5-A80F-E466E1D19AEB),
    helpstring("GohubComPiPackage Class")
]
coclass GohubComPiPackage
{
    [default] interface IGohubPiPackage;
};
[
    uuid(B5A1FD2F-3F4E-4acb-B884-A9AC65850DA2),
    helpstring("GohubComPiPackageToEData Class")
]
coclass GohubComPiPackageToEData
{
    [default] interface IGohubPiPackageToEData;
};
[
    uuid(5848DA64-79D3-41C1-B60C-C88D92508B11),
    helpstring("GohubComFdu92 Class")
]
coclass GohubComFdu92
{
    [default] interface IGohubFdu92;
};
[
    uuid(C2878FA7-4528-4C73-9FF2-D0AA15A91895),

```

```

        helpstring("GohubComGu46 Class")
    ]
coclass GohubComGu46
{
    [default] interface IGohubGu46;
};
[
    uuid(33BBBEE8-8FC6-42BD-9213-A0071F0791EC),
    helpstring("GohubComGu45 Class")
]
coclass GohubComGu45
{
    [default] interface IGohubGu45;
};
[
    uuid(BF5FCE75-E9A7-4282-BFB3-7B8AAB339098),
    helpstring("GohubComGu27 Class")
]
coclass GohubComGu27
{
    [default] interface IGohubGu27;
};
[
    uuid(f7da867b-d7fa-4831-8c81-b8d450d34229),
    helpstring("GohubComInfServsDoc Class")
]
coclass GohubComInfServsDoc
{
    [default] interface IGohubInformServicesDocument;
};
[
    uuid(C5C644CF-8353-470f-AFA9-A12F8396EF87),
    helpstring("GohubComDispatchInfo Class")
]
coclass GohubComDispatchInfo
{
    [default] interface IGohubDispatchInfo;
};
};

```