# Использование javascript-виджетов.

## Общее

Карта – модульное javascript-приложение, являющееся частью Подсистемы веб-доступа ГИС версии 5.5, имеющее интерфейс для работы с внешними приложениями. Widget может встраиваться в настольные или веб-приложения как javascript-модуль. Для взаимодействия используется стандартные вызовы javascript-функций.

Часть интерфейсных функций являются синхронными и возвращают результат сразу. Часть функционала может быть длительным по времени, поэтому реализовано асинхронно: функция не возвращает значение, реализована функция обратного вызова, которая вызывается по окончании операции с одним параметром (true или false), указывающим на успешность операции. Для асинхронных операций в настольных приложениях рекомендуется реализовывать события завершения операции.

## Интерфейс

### Общие функции

#### Общее описание

Для работы с виджетом необходимо загрузить HTML-страницу, содержащую в себе необходимые javascipt-виджеты. Загрузка страницы должна производиться по URL определенного формата. Пример URL для вызова страницы с виджетом карты:

[http://192.168.1.47/web55/main.html?task= MAIN\_WIDGET&mdl=Public/SWF/Map\_55.swf&ll=34.6,43.5&z=19&mapId=MapGenTiles&lineId=1300005&beginKm=0&endKm=100&objects='PODS\_ROUTE:130001,130004|PODS\_ANODE:134501,134502,134503'&select=true&centering=false&showSearch=address,pipe,obj#login=user&UID=0fa9fc06-fe64-4b72-89b8-7ce54d0f7e1d&chk=5b28b8b11a79dd68dd32a8e95f7b1469&ID=1](http://192.168.1.47/web55/main.html?task=%20MAIN_WIDGET&mdl=Public/SWF/Map_55.swf&ll=34.6,43.5&z=19&mapId=MapGenTiles&lineId=1300005&beginKm=0&endKm=100&objects='PODS_ROUTE:130001,130004|PODS_ANODE:134501,134502,134503'&select=true&centering=false&showSearch=address,pipe,obj#login=user&UID=0fa9fc06-fe64-4b72-89b8-7ce54d0f7e1d&chk=5b28b8b11a79dd68dd32a8e95f7b1469&ID=1)

В этом URL базовая часть - <http://192.168.1.47/web55/> это адрес сайта ГИС. Далее идет «main.html» – это html-страница, которая является контейнером для javascript-виджетов. После идет символ «?», после которого идут разделенные символом «&» параметры, необходимые для загрузки виджетов:

* task – параметр, указывающий имя xml-файла с загружаемой задачей. Он используется вместо прежнего метода task(taskName) и позволяет регулировать состав задачи правкой файла задачи. Имя должно быть из набора xml-файлов в папке \WebSite\Core\UITasks\ (т.е. из имеющегося списка задач). Для виджета карты будет использована задача MAIN\_WIDGET.
* mdl – параметр, указывающий относительный путь к swf-модулю карты (если он используется) . Если же swf-модули на html-странице не используются, тогда этот параметр нужно исключить из URL.
* Далее идут необязательные параметры, которые могут использоваться виджетами. В таблице приведены используемые каждым модулем параметры. Отмечены обязательные и необязательные параметры.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Задача** | **Имя параметра** | **Описание параметра** |
| **1** | **MAP (карта)** | ll | x, y – координаты центра карты |
| z | Номер масштаба (0 – весь мир видно, 21 – уровень дома) |
| mapId | Идентификатор карты (по-умолчанию, первая из списка карт в baseLayers.xml) |
| objects=LAYER:ids|LAYER1:ids | Объекты для выделения, центровки. Внимание! Слои могут быть как растровыми, так и векторными. |
| select | Флаг выделения объектов (имеет значение только при передаче параметра objects). Значения true|false |
| centering | Флаг центровки по объектам (имеет значение только при передаче параметра objects). Значения true|false |
| lineId | Идентификатор трубопровода |
| beginKM | Начальный интересующий километр газопровода |
| endKm | Конечный интересующий километр трубопровода |
| showSearch | Отображение панели поиска **address|pipe|obj** |
| **2** | **GRID (таблица)** | template | Имя шаблона (обязательный) |
| params | Условия фильтрации |
| **3** | **CAS (ПКА)** | lineId | Идентификатор трубопровода |
| beginKM | Начальный километр просмотра (если lineId не указан - игнорируется. Если меньше реального начального километра, тоже игнорируется. Если больше конечного километра, то игнорируется) |
| endKm | Конечный километр просмотра (если lineId не указан - игнорируется. Если больше реального конечного километра, тоже игнорируется. Если значение меньше начального километра, то игнорируется) |

* После перечисления всех параметров идет символ «#», за которым следует строка авторизации. Она заменяет прежние функции авторизации – приложение загрузится сразу авторизованным. Состоит из параметров login (имя пользователя), UID (возвращается после однократного логина, сохраняется в cookies при использовании сайта через браузер), chk (специальная проверочная строка, сохраняется в cookies при использовании сайта через браузер), ID (идентификатор пользователя в системе ГИС). Для работы с виджетом карты можно использовать следующую строку: «#login=user&UID=0fa9fc06-fe64-4b72-89b8-7ce54d0f7e1d&chk=5b28b8b11a79dd68dd32a8e95f7b1469&ID=1».

При использовании в настольном приложении контейнера для отображения веб-страниц нужно загрузить страницу по сформированному URL. Для C# приложения это компонент WebBrowser, который загружает страницу методом .Navigate(new Uri(<URL>)). Далее необходимо взаимодействовать с javascipt-функциями загруженной страницы. Описание функций взаимодействия и их параметров не изменилось по сравнению с флеш-версией. Сам способ вызова будет другой и зависит от компонента – вызов из C#-приложения функции в javascript c набором параметров выглядит следующим образом: webBrowser1.Document.InvokeScript(functionName, paramsArr);

вызов внутренней функции C#-приложения из javascript выглядит так:

window.external.SomeMethod('Hello');

пример вызова конкретной функции (на центровке по объекту): webBrowser1.Document.InvokeScript(“findExternalObject”, [“ASMO”,” 17716223985201”]); - будет вызвана центровка на трубопроводе с указанным id.

#### Описание API

### Карта

#### Общее описание

При работе с картой предусмотрены следующие типы операций:

* Поиск объекта на карте с перемещением к нему и выделением. Искать можно, как отдельные объекты (запорную арматуру, КС и др.), так и определенные километры газопровода. (См. ниже описание функций findExternalObject, findExternalPipeKm, findExternalPipePart)
* Нанесение на карту дополнительных слоев данных (например, ВТД), причем как полностью так и частично (отдельные отчеты ВТД, часть оборудования трубопровода). В зависимости от настроек возможна центровка карты к нанесенным данным. (См. ниже описание функций refreshLayer, refreshModel)
* Удаление слоя или группы слоев с карты. (См. ниже описание функций clearModel, clearLayer)
* Получение информации о выделении пользователем объекта или группы объектов. (См. ниже описание функции onObjectsInfo)
* Получение информации о текущем положении на карте: координаты центра карты, ширина и высота карты в километрах и в градусах. (См. ниже описание функций getMapWidth, getMapCenter, getMapPosition).

#### Описание API

* findExternalObject(systemId, objectId) – поиск объекта по идентификаторам системы и объекта. Возвращаемого значения нет. Асинхронная. Обратный вызов findExternalObject (success), где success указывает на успешность операции. Аргументы:
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate)
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate)
* findExternalPipeKm(systemId, objected, km) – поиск километра на газопроводе по идентификаторам системы и объекта и километру. Возвращаемого значения нет. Асинхронная. Обратный вызов findExternalPipeKm (success), где success указывает на успешность операции. Аргументы:
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate)
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate)
  + km – неотрицательное число – километр
* findExternalPipePart (systemId, objected, km, endKm) – поиск участка газопровода по идентификаторам системы и объекта и километровому интервалу. Возвращаемого значения нет. Асинхронная. Обратный вызов findExternalPipePart (success), где success указывает на успешность операции. Аргументы:
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate)
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate)
  + km – неотрицательное число – начальный километр участка
  + endkm – неотрицательное число – конечный километр участка
* getMapWidth() – получение ширины карты в километрах. Синхронная операция, возвращает число или null. Без аргументов.
* getMapCenter() – получение координаты центра карты. Синхронная операция, возвращает массив из двух чисел или null. Координаты в WGS84. Без аргументов.
* getMapPosition() – получение положения карты в формате «ll=*широта*,*долгота*&spn=*ширина\_карты\_в\_градусах*,*высота\_карты\_в\_градусах*&z=*номер\_масштаба*». Синхронная операция, возвращает строку указанного формата или null. Координаты в WGS84. Без аргументов.
* refreshLayer(layerName, params) – обновление данных слоя. Можно использовать для перезагрузки данных слоя или для ограничения нанесенных данных. Список слоев с параметрами будет приведен ниже.
  + layerName – строка - системное имя слоя
  + params – XML с параметрами вида <data ILI\_INSPECTION\_ID=’123’/>. Если во входящих параметрах указаны параметры SYSTEM\_ID, OBJ\_ID, то сначала будет сделан запрос на определение ключевого поля и его значения из таблицы GIS\_INTEGRATION50.GATE и после получения результата подставлены в запрос обновления данных в формате <data {KEY\_FIELDS}={KEY\_VALUES}/>
* clearLayer(layerName) – очистка слоя.
  + layerName – строка - системное имя слоя
* refreshModel(modelName, params) – обновление данных слоев модели (группы слоев)
  + modelName – строка -системное имя модели
  + params – XML с параметрами вида <data ILI\_INSPECTION\_ID=’123’/>
* clearModel(modelName) – очистка данных слоев модели (группы слоев)
  + modelName – строка -системное имя модели
* refreshMap(modelName, params) – обновление данных слоев карты (группы слоев)
  + modelName – строка -системное имя карты
  + params – XML с параметрами вида <data ILI\_INSPECTION\_ID=’123’/>
* clearMap(modelName) – очистка данных слоев карты (группы слоев)
  + modelName – строка -системное имя карты
* onObjectsInfo(objectsList) – функция обратного вызова при запросе пользователя информации по объектам карты.
  + objectsList – XML – XML строка формата <data x="долгота" y="широта" screen\_x="экранная координата x" screen\_y="экранная координата у"> <item id="..." cls="..."/> </data>, где x-долгота точки получения информации, y-широта точки получения информации, id – внутренний идентификатор объекта в ГИС, cls – класс объекта в ГИС. Если объект связан с объектами из ССД Инфотех, то их идентификаторы можно получить запросом: select obj\_id from gis\_integration50.gate where table\_name = '{cls}' and key\_values = '{id}';

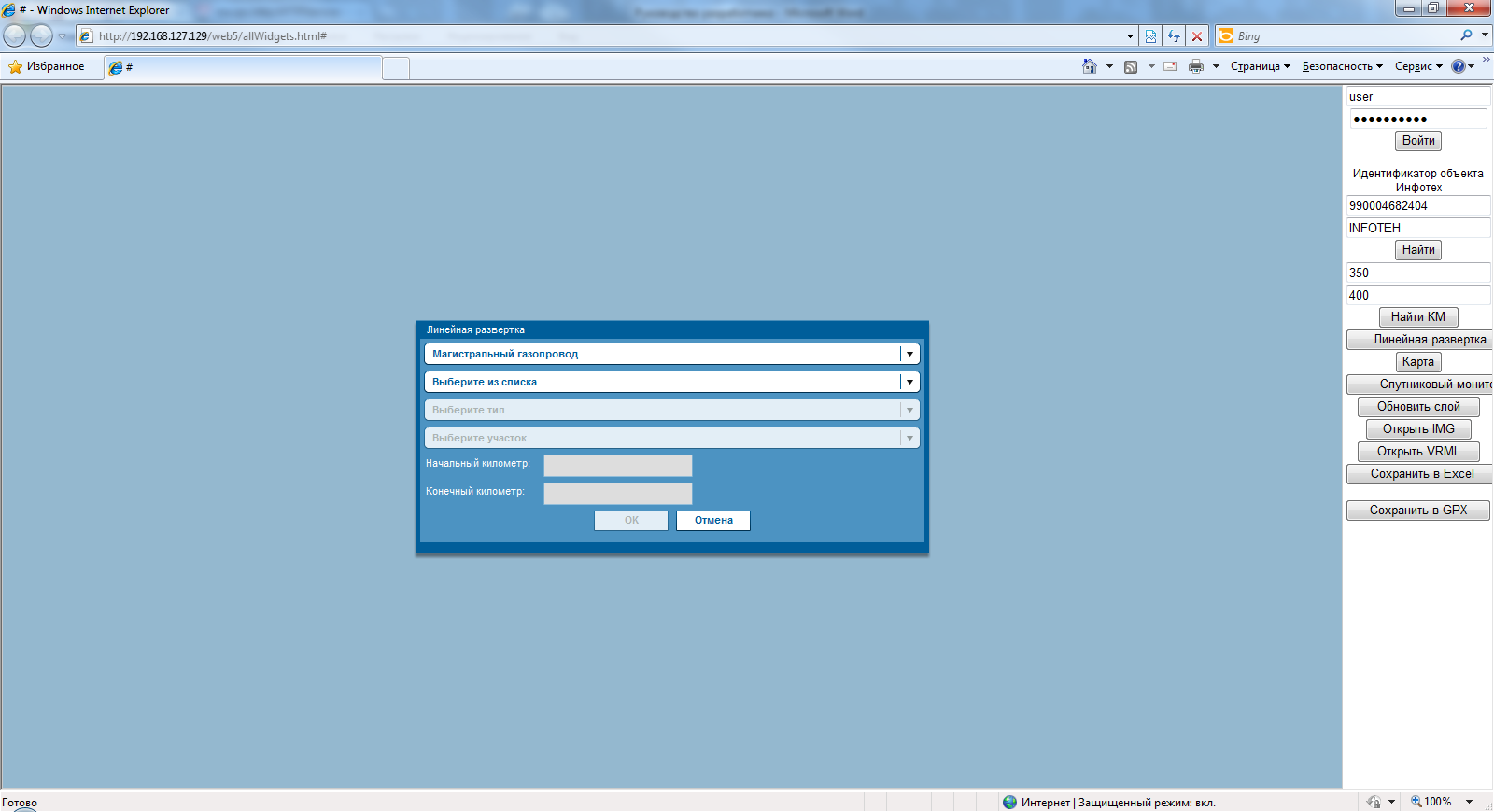
### Подсистема линейной развертки

#### Общее описание

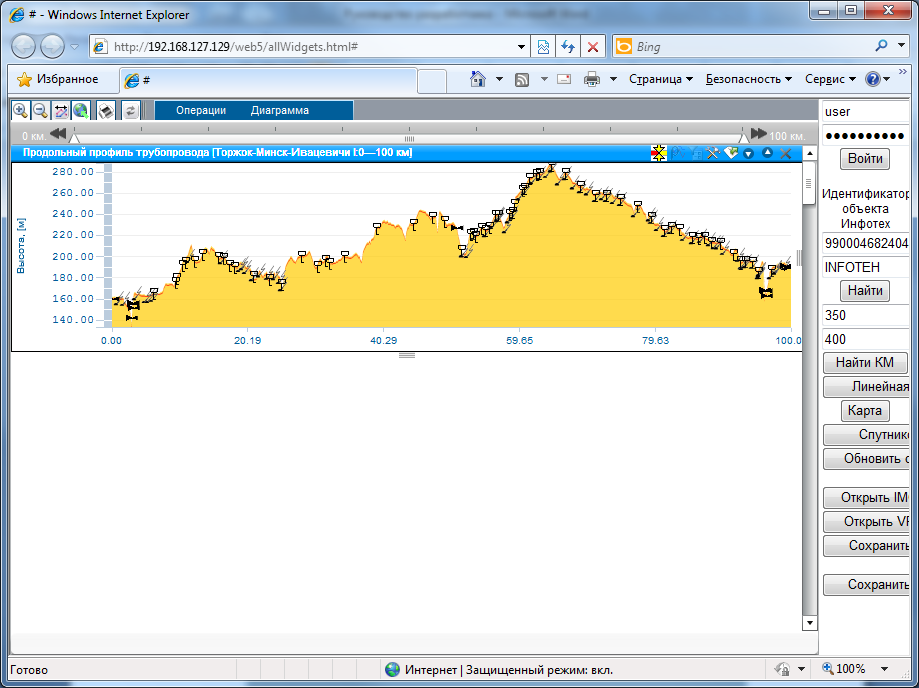
Вызов комплексного анализа доступен через виджет. Для вызова этой задачи нужно в URL изменить параметр task на «CAS».

Возможно два варианта работы: при вызове сразу указать объект анализа (параметрами в URL или через последующую функцию взаимодействия), либо предоставить пользователю возможность выбора трубопровода непосредственно в интерфейсе подсистемы линейной развертки.

Меню выбора объекта выглядит аналогично приведенному ниже рисунку.



Если объект указан при вызове, то это меню не отображается, а сразу отображается профиль указанного трубопровода.



После загрузки данных функциональность полностью совпадает с возможностями подсистемы в основном приложении: просмотр и анализ отчетов ВТД, ЭХЗ средствами порогового, корреляционного анализа, сопоставление расположения объектов на линейной развертке, печать, экспорт в табличный вид (MS Excel).

Описание вызова функции см. в след. разделе.

#### Описание API

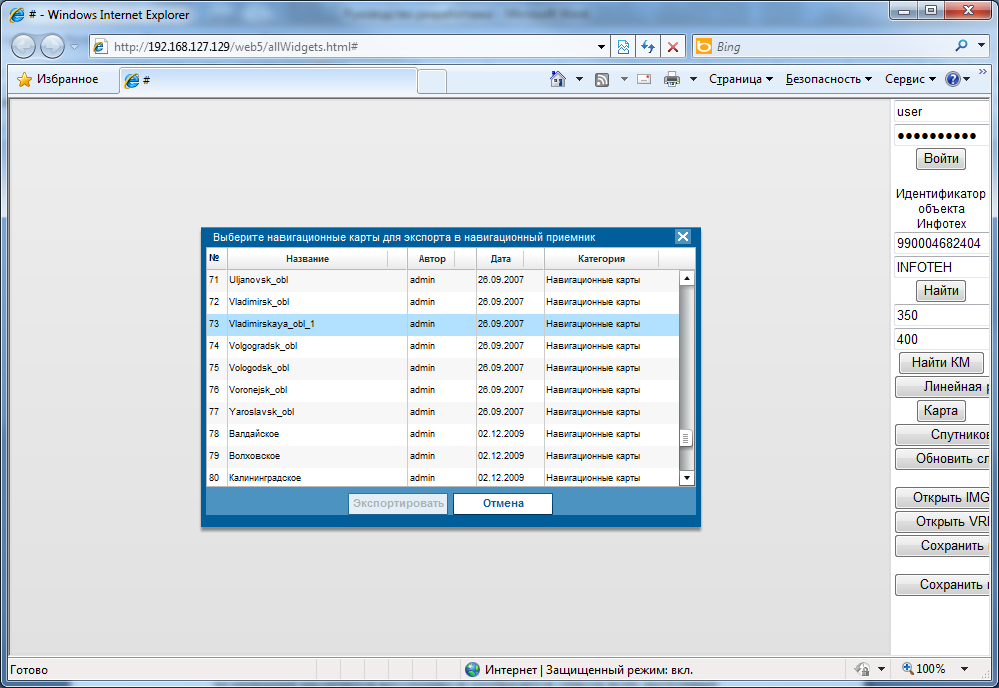
* showCAS(type, systemId, objectId, objectName, kmBegin, kmEnd) – вызов анализа объекта трубопроводной сети. По завершении поиска объекта вызывается функция обратного вызова onshowCAS(success), если объект найден, то вызывается его анализ, а если нет, то вызывается меню выбора объекта анализа.
  + type – строка – тип анализа, допустимые значения: «MG» (магистральный трубопровод), «ROUTE» (участок трубопровода),«CORIDOR» (газотранспортный коридор)
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate). Если пустая строка, то считается, что передается внутренний идентификатор.
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate)
  + kmBegin – число – начальный километр анализа. (Если по данным геопозиционирования указанный трубопровод или коридор имеют бόльший начальный километр, чем указано, то анализ будет начат с фактического начального километра, если параметр kmEnd имеет значение, больше, чем фактический начальный километр.)
  + kmEnd – число – конечный километр анализа. (Если по данным геопозиционирования указанный трубопровод или коридор имеют меньший конечный километр, чем указано, то анализ будет закончен фактическим конечным километром)

### Выгрузка навигационных карт

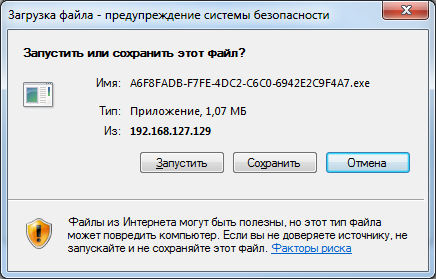
#### Общее описание

Для вызова этой задачи нужно в URL изменить параметр task на «EXPORT\_IMG».

В ГИС содержится большое количество навигационных карт для GPS приемников Garmin (формат IMG) как общие по территории России, так и специфичные для газотранспортного предприятия. Можно выгрузить любое количество имеющихся карт. При вызове отображается меню, в котором можно выбрать нужные карты (см. рисунок ниже).



Если необходимо быстро получить доступ к карте по ЛПУ, то можно при вызове передать необходимые идентификаторы – автоматически будет выбрана карта по ЛПУ и предложено загрузить ее на компьютер – возникнет сообщение, аналогичное приведенному ниже (внешний вид зависит от операционной системы). В этом случае задача может вообще не иметь внешнего интерфейса.



Загружаемый файл – полный набор программ, который необходим для загрузки карты в приемник и карта. Для загрузки карты в устройство необходимо подключить навигатор к компьютеру, запустить загруженный файл и дождаться сообщения об окончании загрузки.

#### Описание API

* exportIMG(systemId, objected) – выгрузка карт для навигационных приемников, связанных с указанным ЛПУ. По завершении работы вызывается функция onexportIMG(success). Если для указанного объекта связанные карты не будут найдены, то операция закончится неуспешно и отобразится список всех доступных навигационных карт.
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate). Если пустая строка, то считается, что передается внутренний идентификатор.
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate)

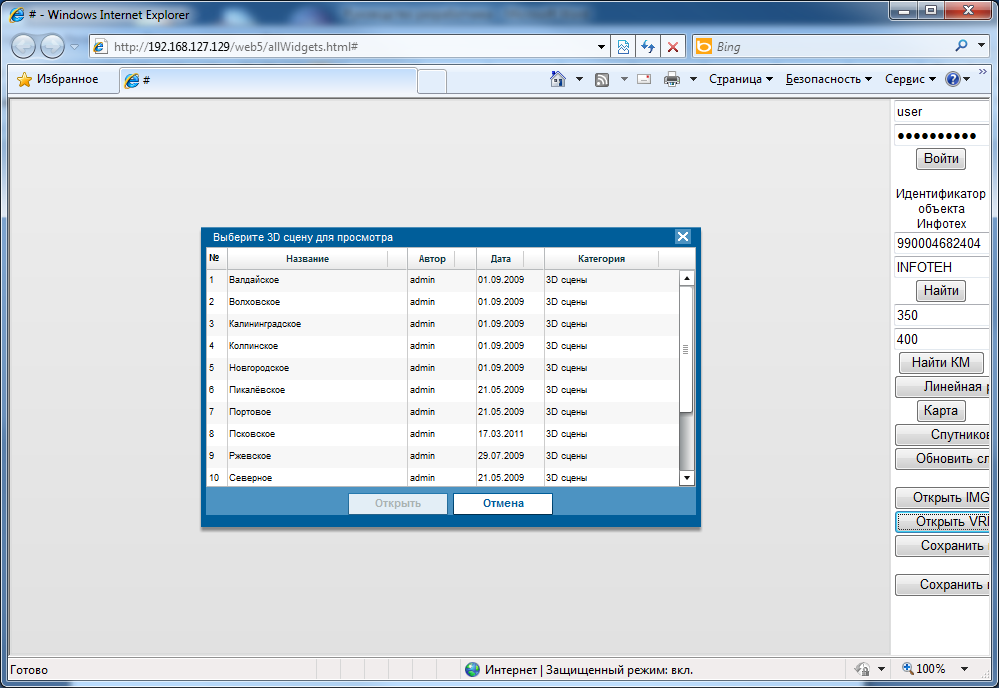
### Выгрузка трехмерных сцен

#### Общее описание

Для вызова этой задачи нужно в URL изменить параметр task на «EXPORT\_WRL».

В ГИС содержатся трехмерные сцены в формате VRML для всех ЛПУ предприятия. Аналогично предыдущим задачам можно вызвать функционал и предоставить пользователю возможность выбрать трехмерную сцену самому (примерный внешний вид показан на рис. ниже) или передать идентификатор ЛПУ при вызове задачи. При явной передачи номера ЛПУ задача тоже может не иметь пользовательского интерфейса.

В обоих случаях после выбора ЛПУ открывается новая страница браузера с выбранной сценой. Если на компьютере не установлено программное обеспечение для просмотра VRML сцен, то будет предложено его установить.



#### Описание API

* openVRML(systemId, objected) – открытие трехмерной сцены по указанному ЛПУ. По завершении работы вызывается функция onopenVRML (success). Если для указанного объекта не будет найдена сцена, то операция закончится неуспешно и отобразится полный список сцен.
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate). Если пустая строка, то считается, что передается внутренний идентификатор.
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate)

### Получение GPX-файла с координатами объектов

#### Общее описание

Для вызова этой задачи нужно в URL изменить параметр task на «EXPORT\_GPX».

В ГИС предусмотрен функционал, позволяющий получить файл с координатами объектов в формате GPX (<http://www.topografix.com/gpx.asp>) для загрузки данных в навигационные устройства. Задача не имеет пользовательского интерфейса, после формирования файла пользователю будет предложено сохранить файл.

Внимание! Для загрузки данных из файла в навигационное устройство необходимо использовать ПО, рекомендованное производителем для загрузки данных.

#### Описание API

* exportGPX(systemId, objected) – получение GPX файла с координатами указанного объекта. Точечные объекты конвертируются к путевый точкам, имеющим называние указанного объекта. Линейные объекты (трубопроводы) – в треки. Площадные объекты – в путевые точки с координатами в центре указанного объекта. По завершении работы вызывается функция onexportGPX(success, params). В параметре params указывается причина ошибки, если операция завершилась неуспешно или путь к gpx-файлу при успешном завершении.  
  ВНИМАНИЕ! Трубопроводы содержат себе большое количество точек, поэтому не все навигационные приемники способны сохранить трек полностью. Необходимо проверять полноту данных после загрузки в приемник.
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate). Если пустая строка, то считается, что передается внутренний идентификатор.
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate)

### Получение Excel-файла с описанием объекта

#### Общее описание

Для вызова этой задачи нужно в URL изменить параметр task на «EXPORT\_EXCEL».

В ГИС предусмотрен функционал, позволяющий получить описание объекта в excel-формате. Задача не имеет пользовательского интерфейса, после формирования файла пользователю будет предложено сохранить файл.

#### Описание API

* exportExcel(systemId, objected) – получение Excel-файла с описанием объекта.
  + systemId – строка – идентификатор системы. (Значение из колонки system\_id таблицы gis\_integration50.gate). Если пустая строка, то считается, что передается внутренний идентификатор.
  + objectId – строка – идентификатор объекта. (Значение из колонки obj\_id таблицы gis\_integration50.gate, если идентификатор из внешней системы)

## Порядок работы

1. Добавление WebBrowser-контрола в приложение и загрузка URL. Сразу авторизуемся через строку авторизации в URL.
2. Ожидание события окончания загрузки.
3. Ожидаем окончания загрузки карты (либо другого виджета). По окончании загрузки вызывается функция *mapReady(true)* и *taskReady(“mapReady”, true)* .
4. После этого можно работать с функционалом поиска объектов, получения центра и положения карты.

## Примеры

### С#

В комплекте документации приложен архив проекта на C#, демонстрирующий работу с widget (находится в архиве JS*WidgetExample.zip*). Работоспособность примера проверена на Microsoft Visual Studio версии 10 и Flash Player версии 10 (браузер InternetExplorer10). На более поздних версия указанных программ возможна необходимость корректировки кода подключения Flash-компонента.

Адрес сайта и вынесен в поле «URL-адрес загружаемой страницы», кнопка «Загрузить страницу» вызовет загрузку сайта внутри компонента WebBrowser. После полной загрузки сайта можно указывать любую из перечисленных функций взаимодействия с загруженным виджетом, добавляя параметры в табличку с одной колонкой «параметры».

Основные моменты:

1. Form1.cs – обертка для работы с html-страницами с widget-ами
2. Для работы по загрузке сайта приложением используется компонент WebBrowser.
3. Обязательно нужно дождаться окончания загрузки страницы: события WebBrowser’a DocumentCompleted. Далее нужно ждать вызова функции готовности виджета (например mapReady() и taskReady() для карты). Вызовы javascript-методов widget выполняются вызовом метода webBrowser1.Document.InvokeScript(functionName, paramsArr);
4. Обратные вызовы от widget приходят в специальный класс ScriptManager. Пример вызова в функции-конструкторе визуальной формы public Form1().

### HTML + JavaScript

В поставку системы входит страница main.html, демонстрирующий весь функционал сайта. Полностью сайт загружается по ссылке http:\\<адрес сайта>\web55\main.html. Компонент WebBrowser реализует возможность взаимного вызова функций между сайтом и настольным приложением. Иначе вызывать javascript-методы страницы main.html и виджетов не получится (по соображениям безопасности InternetExplorer вызов js-функций загруженной страницы извне невозможен). Для Delphi альтернативой C#-компонента WebBrowser является TWebBrowser, и судф по описанию его функционал аналогичен.