



**Руководство разработчика по
работе с системой AMBER**

Оглавление

1.	Описание системы	4
2.	Вход в систему.....	5
3.	Интерфейс системы	7
4.	Описание основных модулей	9
5.	Сценарии работы в системе	10
5.1.	Настройка отчетов системы.....	10
5.1.1.	Описание.....	10
5.1.2.	Добавление нового отчета	10
5.1.2.1.	Создание представления в системе.....	10
5.1.2.2.	Создание шаблона в Excel	11
5.1.2.3.	Регистрация отчета в системе.....	11
5.1.3.	Редактирование отчета в системе	13
5.1.4.	Удаление существующего отчета	14
5.1.5.	Пример использования	14
5.2.	Настройка печатных форм.....	17
5.2.1.	Описание.....	17
5.2.2.	Добавление новой печатной формы.....	17
5.2.2.1.	Создание представления в системе.....	17
5.2.2.2.	Создание Word шаблона.....	17
5.2.2.3.	Регистрация отчета в системе.....	18
5.2.3.	Редактирование печатной формы в системе	20
5.2.4.	Удаление существующей печатной формы	21
5.2.5.	Пример использования	22
6.	Описание API системы	26
6.1.	Описание свойств объектов структуры	26
6.1.1.	Объект Structure	26
6.1.2.	Объект Property	26
6.1.3.	Значения справочника Property types:	27
6.1.4.	Объект ClassReference	27
6.1.5.	Объект Localization	27

6.1.6.	Объект SyntheticProperty	28
6.1.7.	Объект RuleValue	28
6.1.8.	Объект Numerator	28
6.1.9.	Объект NumeratorArgument	28
6.2.	Аутентификация.....	29
6.3.	Получение информации по объекту	30
6.4.	Создание записи данных объекта	33
6.5.	Редактирование записи данных объекта.....	34
6.6.	Удаление записи данных объекта	35
6.7.	Получение списка записей данных объекта с возможностью фильтрации и пейджинга	35
6.8.	Получение записи по идентификатору	38
6.9.	Получение данных объекта по «расширенному» запросу.....	40

1. Описание системы

Amber - программная платформа, предназначенная для поддержки роста продаж и роста эффективности бизнеса.

Возможности продукта:

- Формирование базы потенциальных клиентов для активных продаж с учетом индивидуальных особенностей бизнеса;
- Встроенный эталонный процесс продаж и процесс обработки входящих запросов;
- Интеграция системы со всеми источниками поступления обращений в компанию: сайт, телефония, почта, онлайн-сервисы по работе с клиентами;
- Мощная аналитика по продажам в любых срезах и эффективности работы

менеджеров. Amber помогает решать задачи:

- поиска клиентов;
- централизованного управления клиентами, контактами, сделками;
- эффективная отработка 100% запросов, поступающих в компанию;
- повышение дохода с клиентов, суммы среднего чека и объема повторных продаж;
- поиск проблемных мест в продажах и сокращение число отказов от сделок;
- повышение результативности менеджеров, работающих с клиентами;
- контроль продаж и выполнение планов по направлениям, регионам, менеджерам, подразделениям;
- планирование и прогноз продаж на основе статистики, трендов, незавершенных сделок;
- постановка задачи сотрудникам и контролировать сроки их выполнения;
- оценки эффективности маркетинговых инструментов.

2. Вход в систему

Для входа в систему пользователь должен ввести в адресную строку браузера адрес системы Amber, используемой в его компании. В качестве браузера рекомендуется использовать Google Chrome, Chromium, Yandex.Браузер и аналоги.

Система Amber реализует 3 вида авторизации пользователей:

- Авторизация в системе по связке логин-пароль;
- Авторизация в системе с помощью Active Directory (возможна только при использовании в компании системы доменных имен и настроенной интеграции с ней);
- Смешанная авторизация (когда пользователь может авторизоваться либо первым, либо вторым способом).

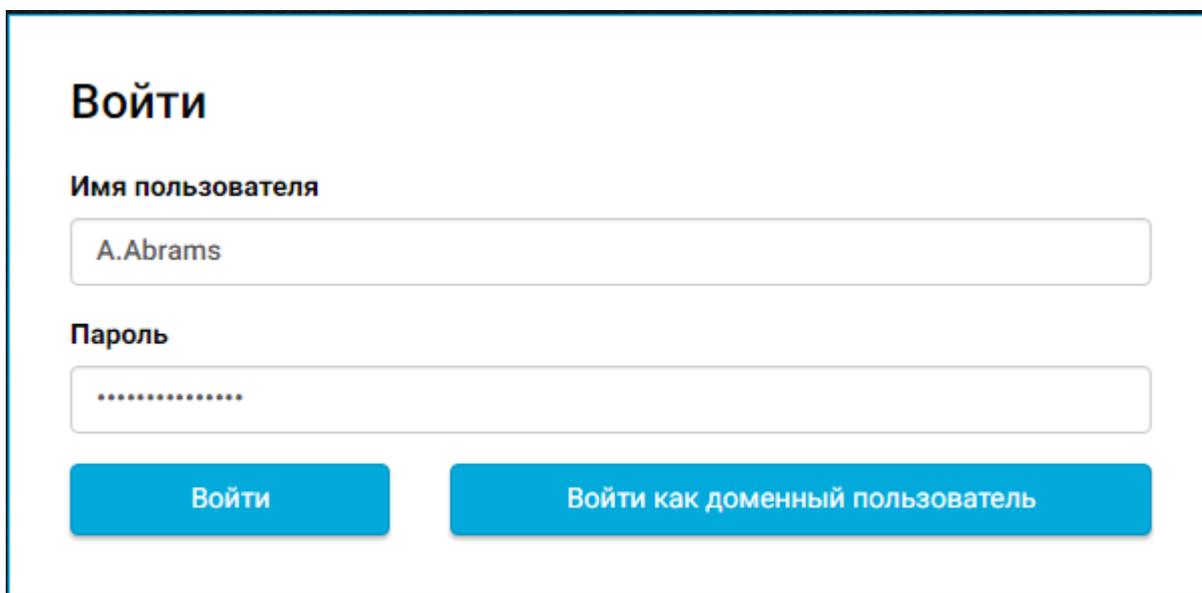


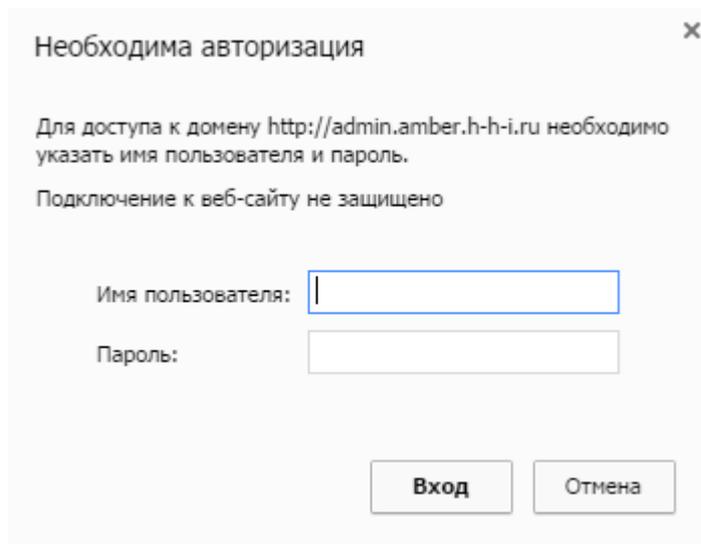
Рисунок 1. Окно авторизации пользователя в Системе

Для авторизации с помощью имени пользователя и пароля пользователь должен ввести свои учетные данные в соответствующие поля и нажать кнопку **Войти**.

Для авторизации с помощью Active Directory, пользователь просто должен нажать кнопку **Войти как доменный пользователь**. Данный вид авторизации доступен при следующих условиях:

- В системе используется доменная система Active Directory;
- Настроена интеграция Amber с Active Directory.

Если пользователь авторизуется в системе «извне» (не из внутренней сети компании), то для доменной авторизации также придется ввести имя пользователя (доменного) и пароль.



Необходима авторизация

Для доступа к домену <http://admin.amber.h-h-i.ru> необходимо указать имя пользователя и пароль.

Подключение к веб-сайту не защищено

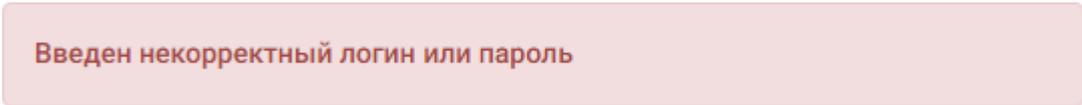
Имя пользователя:

Пароль:

Рисунок 2. Доменная авторизация в системе

Смешанная авторизация подразумевает возможность авторизации пользователя как через доменную авторизацию, так и через авторизацию по логину-паролю.

Если после авторизации отображается ошибка «Введен некорректный логин или пароль»,



Введен некорректный логин или пароль

Рисунок 3. Ошибка авторизации

то необходимо проверить введенные имя пользователя и пароль и попробовать авторизоваться в системе повторно. Если ошибка повторяется, то обратитесь к администратору системы.

3. Интерфейс системы

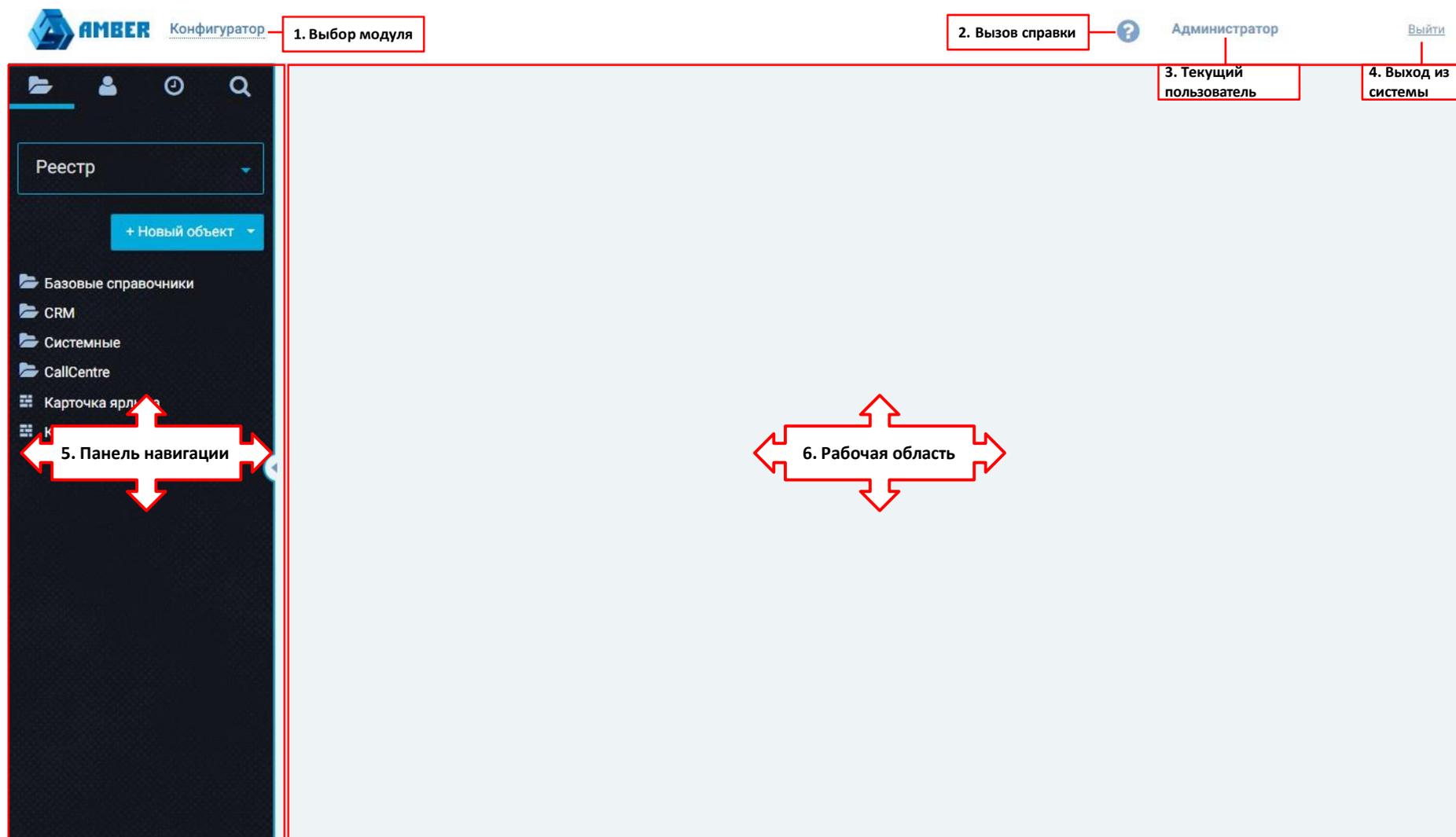


Рисунок 4. Интерфейс системы для работы администратора

В интерфейсе доступны следующие элементы:

1. **Выбор модуля** – кнопка выбора одного из 6 модулей системы для настройки системы.

Для выбора модуля необходимо нажать на название активного модуля, после чего откроется список доступных модулей:

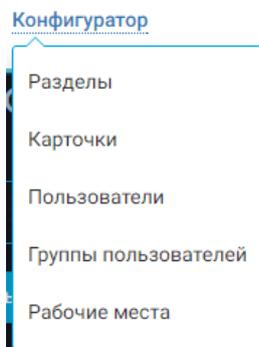


Рисунок 5. Список модулей

Для перехода к какому-либо модулю пользователю необходимо выбрать соответствующую строку и нажать по ней левой клавишей мыши. Модули подробно описаны в разделе 3.

2. **Вызов справки** – кнопка для вызова справки по работе в административной части системы. Для вызова справки необходимо нажать на кнопку  ;
3. **Текущий пользователь** – ФИО контакта пользователя, под которым авторизовался администратор. Данный элемент предназначен для информации, он не имеет активных функций;
4. **Выход из системы** – кнопка для возврата на окно ввода логина и пароля. После нажатия на кнопку выхода пользователь деавторизуется в системе и попадет на страницу авторизации (раздел 2);
5. **Панель навигации** – панель для навигации по модулям. Изменяется в зависимости от выбранного модуля;
6. **Рабочая область** - область для работы с объектами модулей. Изменяется в зависимости от выбранного элемента модуля на панели навигации.

4. Описание основных модулей

В UI администратора системы Amber доступны следующие модули:

1. **Конфигуратор** – модуль предназначен для создания и ведения объектов системы и экранных форм для этих объектов.
2. **Разделы** – модуль для создания и ведения разделов системы;
3. **Карточки** – модуль регистрации экранных форм для объектов в качестве форм по умолчанию (в будущем – форм по условиям)
4. **Пользователи** – модуль для создания и ведения пользователей системы (создание, назначение/изменение паролей, удаления, настройки типов авторизации)
5. **Группы пользователей** – модуль для создания и ведения групп пользователей. Группы облегчают назначение пользователям прав т.к. все права, назначенные группе пользователей, распространяются на всех пользователей, состоящих в группе.
6. **Рабочие места** – модуль для создания и ведения рабочих мест пользователей, а также назначения рабочих мест пользователям и группам.
7. **Лицензии** – модуль предназначен для формирования файлов-запросов на получение лицензий.

В каждом модуле все элементы интерфейса, за исключением панели навигации и рабочей области, идентичны. Поэтому в рамках модулей будут рассмотрены функции и особенности панели навигации и рабочей области.

5. Сценарии работы в системе

5.1. Настройка отчетов системы

5.1.1. Описание

В данном разделе описаны сценарии добавления новых отчетов в систему, изменения и удаления существующих отчетов.

5.1.2. Добавление нового отчета

Добавление нового отчета можно разделить на 3 этапа:

1. Создание представления в БД;
2. Создание Excel шаблона;
3. Регистрация отчета в системе.

5.1.2.1. Создание представления в системе

Данный этап может быть выполнен только при наличии доступа к БД системы Amber.

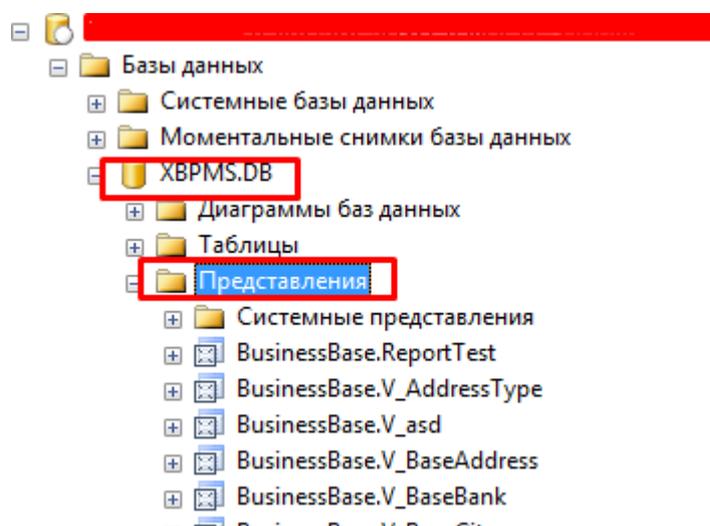
Представление необходимо для формирования таблицы, из значений которой будет строиться отчет.

Первым делом необходимо создать источник данных для отчета – представление в БД. Использование функций СУБД MS SQL для создания представлений подробно расписано в официальной документации Microsoft, доступной по данной [ссылке](#).

Примеры построения представлений приведены в разделе 5.1.5. Важно помнить, что в представлении должна присутствовать часть с разграничением прав пользователей на записи, иначе отчет не будет формироваться.

Для создания представления производим следующие шаги:

- 1) Заходим в Microsoft SQL management studio и подключаемся к СУБД с базой данных



Amber (XBPMS.DB)

Рисунок 6. Подключение к СУБД

- 2) В БД amber создаем представление (логика построения зависит от конкретной задачи)

- 3) Сохраняем представление в формат: [BusinessBase].[V_REPORT_{название представления}]

5.1.2.2. Создание шаблона в Excel

Далее необходимо создать шаблон в MS Excel

- 1) Создаем новый файл Excel;
- 2) В excel файле добавляем таблицу и присваиваем имя такое же как у представления только без BusinessBase (V_REPORT_{название представления});

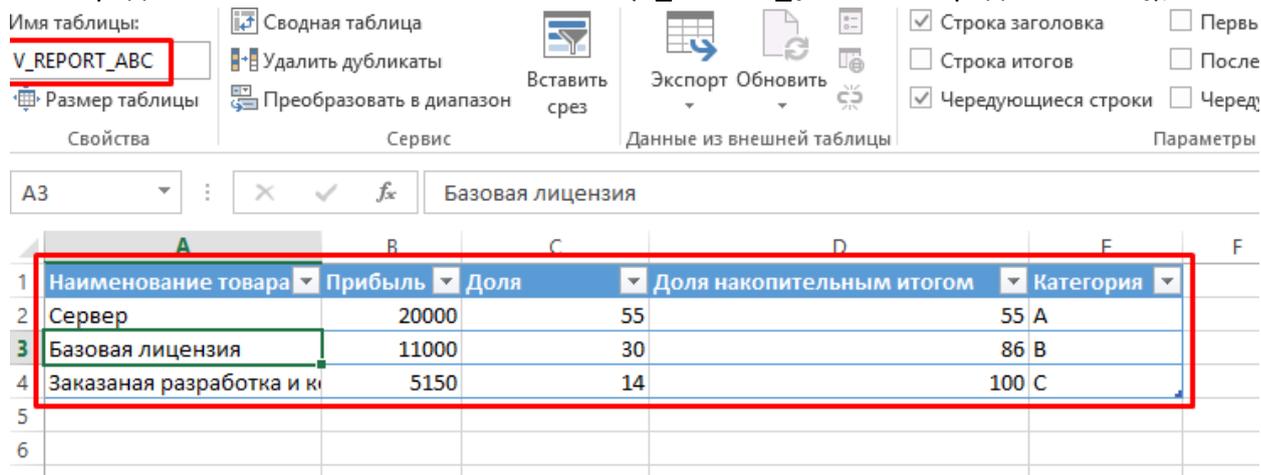


Рисунок 7. Настройка шаблона в Excel

- 3) Проставляем формат колонок. На данный момент обрабатываются только следующие форматы:
 - Числовой (без знаков);
 - Дата (*14.03.2012);
 - Все остальные поля оставляем в Общем формате.
- 4) Настраиваем содержимое шаблона. Самый простой способ использовать таблицу как источник для сводной диаграммы или сводной таблицы.
- 5) Сохраняем файл. Имя файла должно быть написано латиницей (A-z). Формат xlsx.

5.1.2.3. Регистрация отчета в системе

Далее необходимо зарегистрировать отчет в системе Amber.

1. Заходим в пользовательскую часть системы под учетной записью администратора;

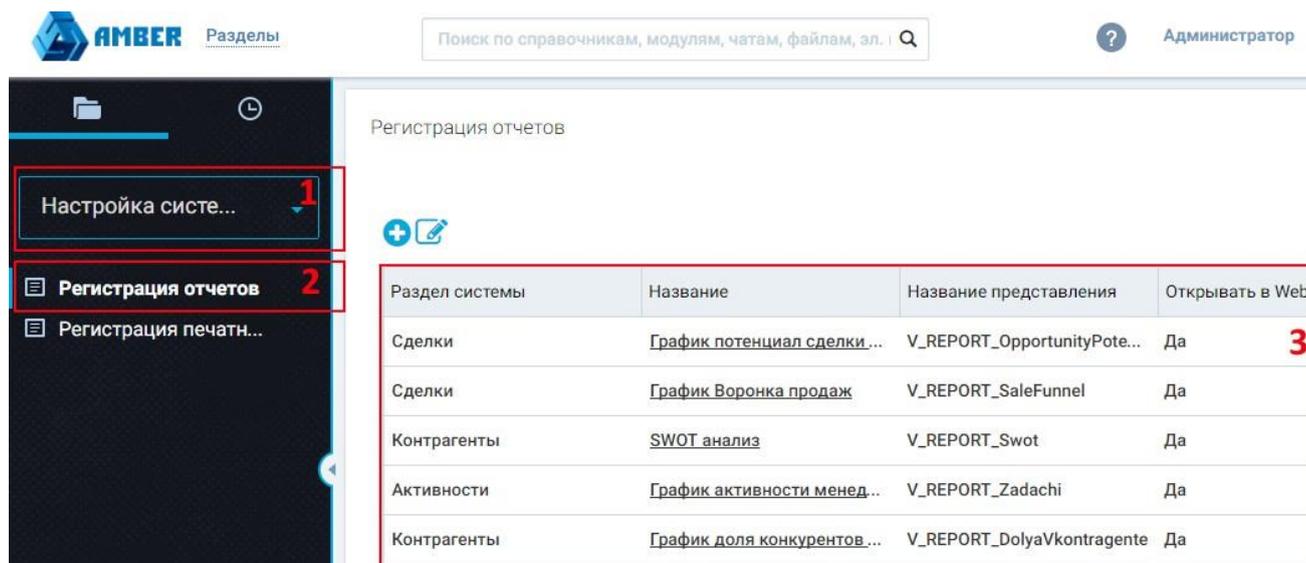


Рисунок 8. Регистрация отчета в Amber

- В списке рабочих мест (1 на рисунке) выбираем «Настройка системы»;
- Выбираем раздел «Регистрация отчетов» (2 на рисунке);
- Для регистрации нового отчета нажимаем **+** (3 на рисунке);
- Откроется карточки регистрации нового отчета;

Регистрация отчетов • Карточка Отчета

Название	Название представления	Раздел системы	Идентификатор отчета
График активности менед...	V_REPORT_Zadachi	Активности	29
ManagerActivity.xlsx ✕			
<input checked="" type="checkbox"/> Открывать в Web			

Рисунок 9. Карточка регистрации отчета

Описание полей карточки:

- **Название** – отображаемое в системе название отчета;
- **Название представления** – указываем ранее созданное представление;
- **Раздел системы** – выбираем в каком разделе системы должен отображаться отчет;
- **Идентификатор отчета** – идентификатор отчета в системе. Если поле идентификатор отчета заполнено, то происходит обновление данных отчета. Если поле идентификатор отчета пустое, то происходит регистрация нового отчета;
- **Область для прикрепления файла** – место для загрузки файла отчета. Можно либо перетянуть файл на эту область, либо нажать и выбрать файл из обозревателя файлов;
- **Открывать в Web** - указываем надо ли открывать отчет в SharePoint, либо необходимо сразу загружать excel файл.

6. Заполняем данные отчета и после завершения настройки отчета нажимаем «Сохранить».

5.1.3. Редактирование отчета в системе

Сценарий редактирования отчетов Amber практически совпадает со сценарием добавления новых отчетов. Отличие в том, что отчет уже существует в системе и возможен выбор в разделе Регистрация отчетов.

1. Заходим в пользовательскую часть системы под учетной записью администратора;

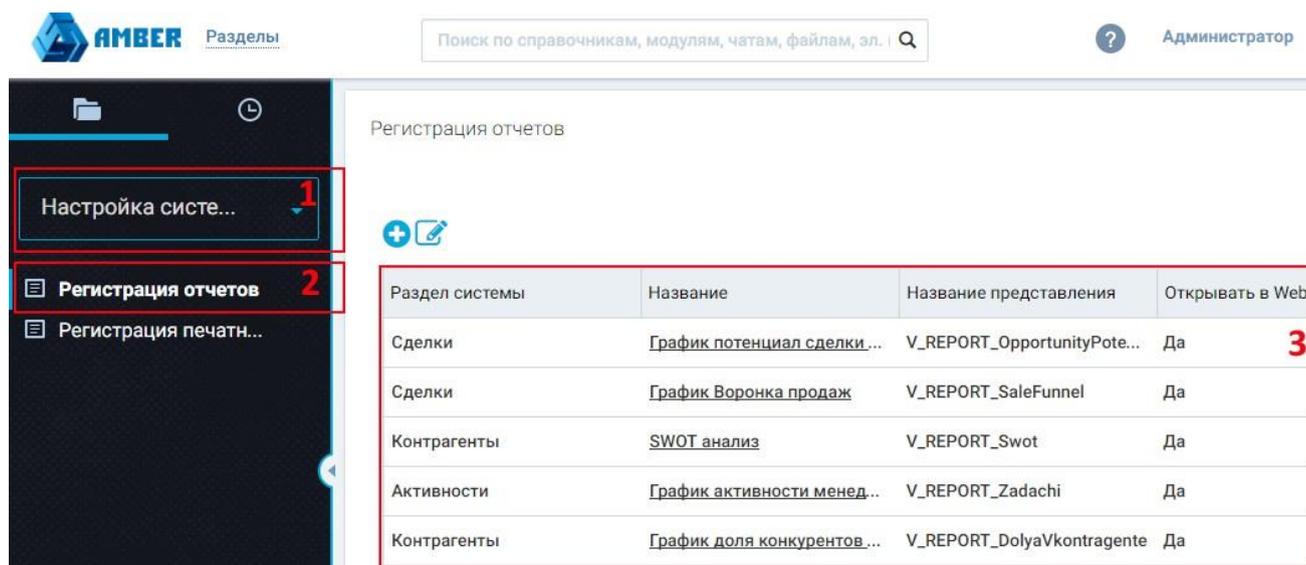


Рисунок 10. Раздел регистрации отчетов

2. В списке рабочих мест (1) выбираем «Настройка системы»;
3. Выбираем раздел «Регистрация отчетов»;
4. Открываем существующий отчет для последующего редактирования;
5. Откроется карточки отчета;

Регистрация отчетов • Карточка Отчета

Название	Название представления	Раздел системы	Идентификатор отчета
График активности менедж...	V_REPORT_Zadachi	Активности	29
ManagerActivity.xlsx ✕			
<input checked="" type="checkbox"/> Открывать в Web			

Рисунок 11. Карточка регистрации отчета

Описание полей карточки:

- **Название** – отображаемое в системе название отчета;
- **Название представления** – указываем ранее созданное представление;
- **Раздел системы** – выбираем в каком разделе системы должен отображаться отчет;
- **Идентификатор отчета** – идентификатор отчета в системе. Если поле идентификатор отчета заполнено, то происходит обновление данных отчета. Если

поле идентификатор отчета пустое, то происходит регистрация нового отчета;

- **Область для прикрепления файла** – место для загрузки файла отчета. Можно либо перетянуть файл на эту область, либо нажать и выбрать файл из обозревателя файлов;
- **Открывать в Web** - указываем надо ли открывать отчет в SharePoint, либо необходимо сразу загружать excel файл.

6. Заполняем данные отчета и после завершения настройки отчета нажимаем «Сохранить».

5.1.4. Удаление существующего отчета

Для удаления отчета производим следующие шаги:

1. Заходим в пользовательскую часть системы под учетной записью администратора;

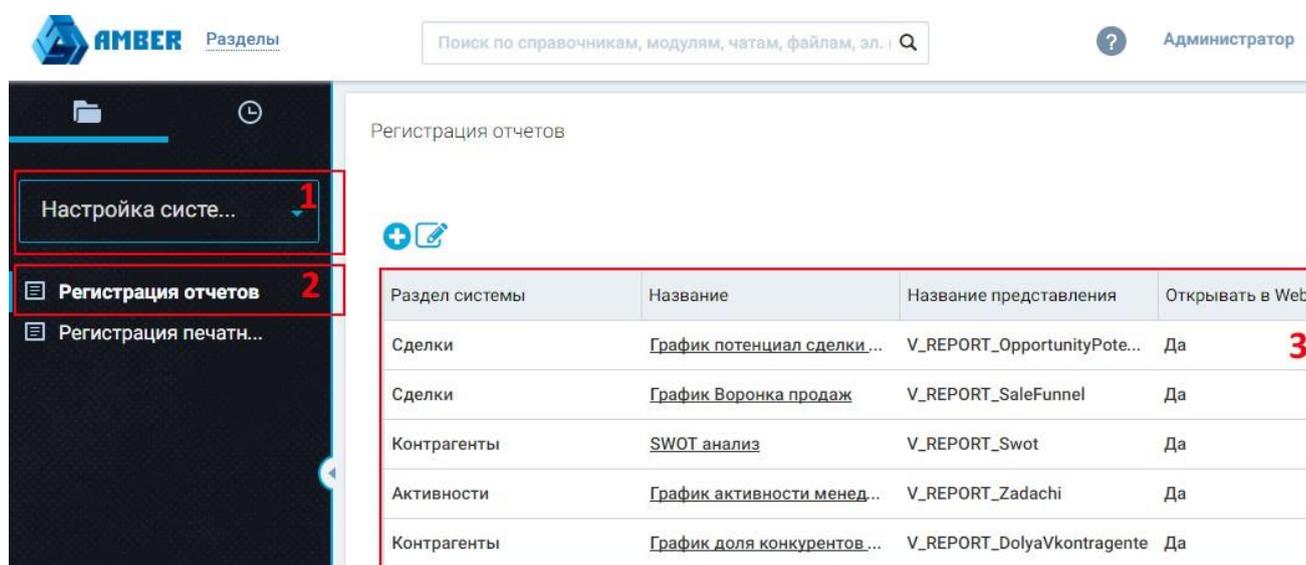


Рисунок 12. Раздел регистрации отчетов

2. В списке рабочих мест (1) выбираем «Настройка системы»;
3. Выбираем раздел «Регистрация отчетов»;
4. Удаляем существующий отчет.

5.1.5. Пример использования

1. Создаем новое представление в БД;

```
select t.[Product], t.ProductId, t.[Variation coefficient, V],
(case
  when t.[Variation coefficient, V] < 10 THEN 'X'
  when t.[Variation coefficient, V] >= 10 AND t.[Variation coefficient, V] <
25 THEN 'Y'
  when t.[Variation coefficient, V] >= 25 then 'Z'
end
) as [Group]
from (
SELECT      bp.Name          as      [Product],          bp.Id          as      ProductId,
CAST(( (STDEVP (pio.SumWitTax) /AVG (pio.SumWitTax) *100) as      Int)      as      [Variation
coefficient, V]
FROM BusinessBase.ProductInDeal AS pid
INNER JOIN BusinessBase.ProductInObject AS pio ON pio.Id = pid.Id
```

```

INNER JOIN BusinessBase.BaseProduct AS bp ON bp.Id = pio.Product
WHERE pio.SumWitTax>0
group by bp.Name, bp.Id
    
```

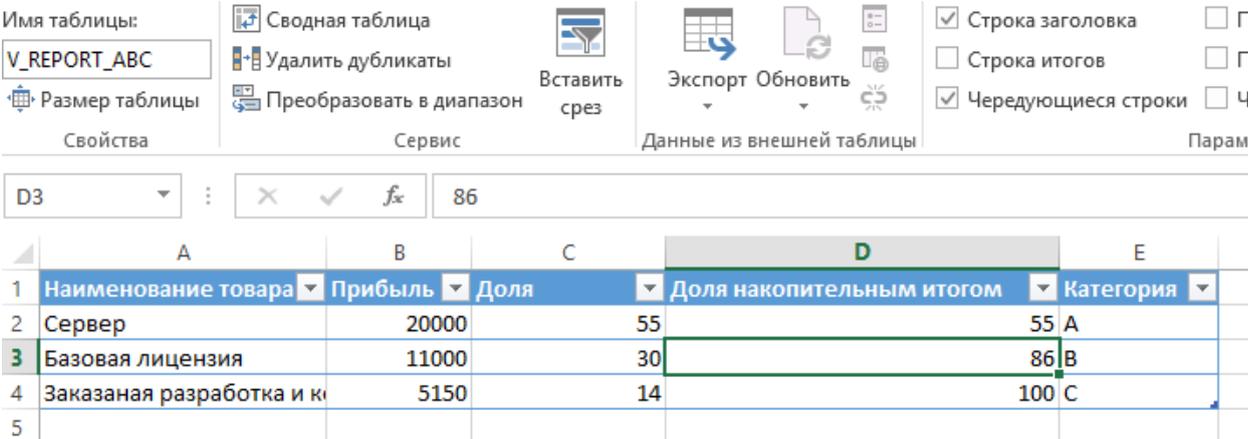
2. В представление добавляем часть для разграничения прав;

```

CREATE VIEW [BusinessBase].[V_REPORT_XYZ] AS
select t2.Product, t2.[Variation coefficient, V], t2.[Group], zz.IdUser as UserId from
(
select t.[Product], t.ProductId, t.[Variation coefficient, V], (case
when t.[Variation coefficient, V] < 10 THEN 'X'
when t.[Variation coefficient, V] >= 10 AND t.[Variation coefficient, V] <
25 THEN 'Y'
when t.[Variation coefficient, V] >= 25 then 'Z'
end
) as [Group] from(
SELECT bp.Name as [Product], bp.Id as ProductId, CAST((STDEVP(pio SumWitTax)/AVG(pio SumWitTax)*100) as Int)
as [Variation coefficient, V]
FROM BusinessBase.ProductInDeal AS pid
INNER JOIN BusinessBase.ProductInObject AS pio ON pio.Id = pid.Id INNER JOIN
BusinessBase.BaseProduct AS bp ON bp.Id = pio.Product
WHERE pio.SumWitTax>0 group by
bp.Name, bp.Id
)t
order by t.[Variation coefficient, V] asc
)t2
INNER JOIN
(
SELECT x.Id AS IdUser, e.id AS IdRecord FROM
Permission.xbpmsUser AS x
CROSS APPLY core.EntitiesForUser(x.id, 'BaseProduct', 1) AS e
) AS ZZ ON zz.IdRecord = t2.ProductId
    
```

3. После создаем новый excel;

4. В экселе создаем таблицу и указываем название таблицы;



1	Наименование товара	Прибыль	Доля	Доля накопительным итогом	Категория
2	Сервер	20000	55	55	A
3	Базовая лицензия	11000	30	86	B
4	Заказанная разработка и к	5150	14	100	C
5					

Рисунок 13. Создание шаблона в Excel

5. В таблице должны присутствовать все поля из представления, кроме UserId. UserId это системное поле необходимое в представлении только для разграничения прав на записи. Так же необходимо учитывать, что названия полей в представлении и в excel файле должны совпадать;
6. Далее регистрируем отчет;
7. Заходим в пользовательскую часть системы под учетной записью администратора;
8. В списке рабочих мест выбираем «Настройка системы»;
9. Выбираем раздел «Регистрация отчетов»;
10. Для регистрации нового отчета нажимаем ;
11. Откроется карточки регистрации нового отчета;

Регистрация отчетов • Карточка Отчета

Название	Название представления	Раздел системы	Идентификатор отчета
График активности менедж	V_REPORT_Zadachi	Активности	29
<div style="border: 1px dashed #00aaff; padding: 5px; display: inline-block;">  ManagerActivity.xlsx × </div>			
<input checked="" type="checkbox"/> Открывать в Web			

Рисунок 14. Карточка регистрации отчета

Описание полей карточки:

- **Название** – отображаемое в системе название отчета;
- **Название представления** – указываем ранее созданное представление;
- **Раздел системы** – выбираем в каком разделе системы должен отображаться отчет;
- **Идентификатор отчета** – идентификатор отчета в системе. Если поле идентификатор отчета заполнено, то происходит обновление данных отчета. Если поле идентификатор отчета пустое, то происходит регистрация нового отчета;
- **Область для прикрепления файла** – место для загрузки файла отчета. Можно либо перетянуть файл на эту область, либо нажать и выбрать файл из обозревателя файлов;
- **Открывать в Web** - указываем надо ли открывать отчет в SharePoint, либо необходимо сразу загружать excel файл.

12. Заполняем данные отчета и после завершения настройки отчета нажимаем «Сохранить».

5.2. Настройка печатных форм

5.2.1. Описание

В данном разделе описаны сценарии по настройке, созданию и удалению печатных форм в системе Amber.

5.2.2. Добавление новой печатной формы

Добавление нового отчета можно разделить на 3 этапа:

1. Создание представления(ий) в БД;
2. Создание Word шаблона;
3. Регистрация печатной формы в системе.

5.2.2.1. Создание представления в системе

Данный этап может быть выполнен только при наличии доступа к БД системы Amber.

Представление необходимо для формирования таблицы, из значений которой будет строиться печатная форма.

Первым делом необходимо создать источник данных для печатной формы – представление в БД. Использование функций СУБД MS SQL для создания представлений подробно расписано в официальной документации Microsoft, доступной по данной [ссылке](#).

Примеры построения представлений приведены в разделе

5.2.5. Для создания представления производим следующие шаги:

1. Заходим в Microsoft SQL management studio

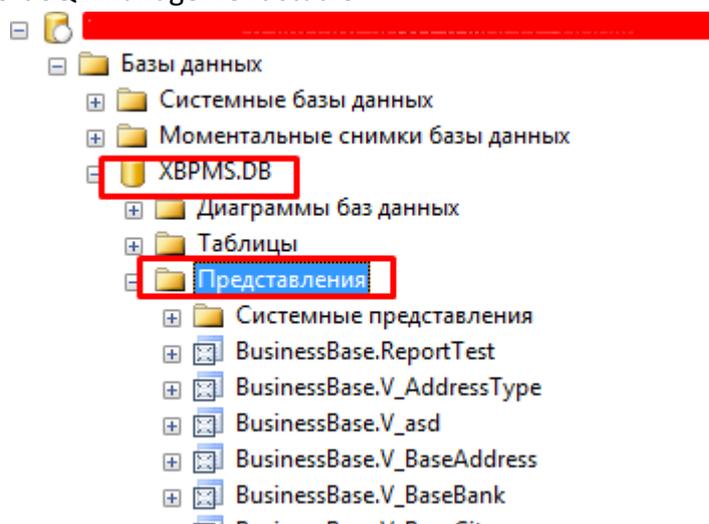


Рисунок 15. Подключение к СУБД

2. В БД amber создаем представление (логика построения зависит от конкретной задачи)
3. Сохраняем представление в формат: [BusinessBase].[V_PR_{название представления}]
4. Создаем дополнительные представления для табличных частей документа (например, продукты в договоре).

5.2.2.2. Создание Word шаблона

Далее необходимо создать шаблон в MS Word.

1. Создаем новый файл MS Word;

- В word файле добавляем текст печатной формы и указываем метки для подстановки данных.
 [[Метка]] – обычная текстовая метка. При формировании заменяется данными из выбранной записи;
 {{Табличная метка}} – табличная метка. При формировании замещается таблицей.
- Сохраняем файл печатной формы. Имя файла должно быть написано латиницей (A-z).
 Формат xlsx.

5.2.2.3. Регистрация отчета в системе

Далее необходимо зарегистрировать отчет в системе Amber.

- Заходим в систему;

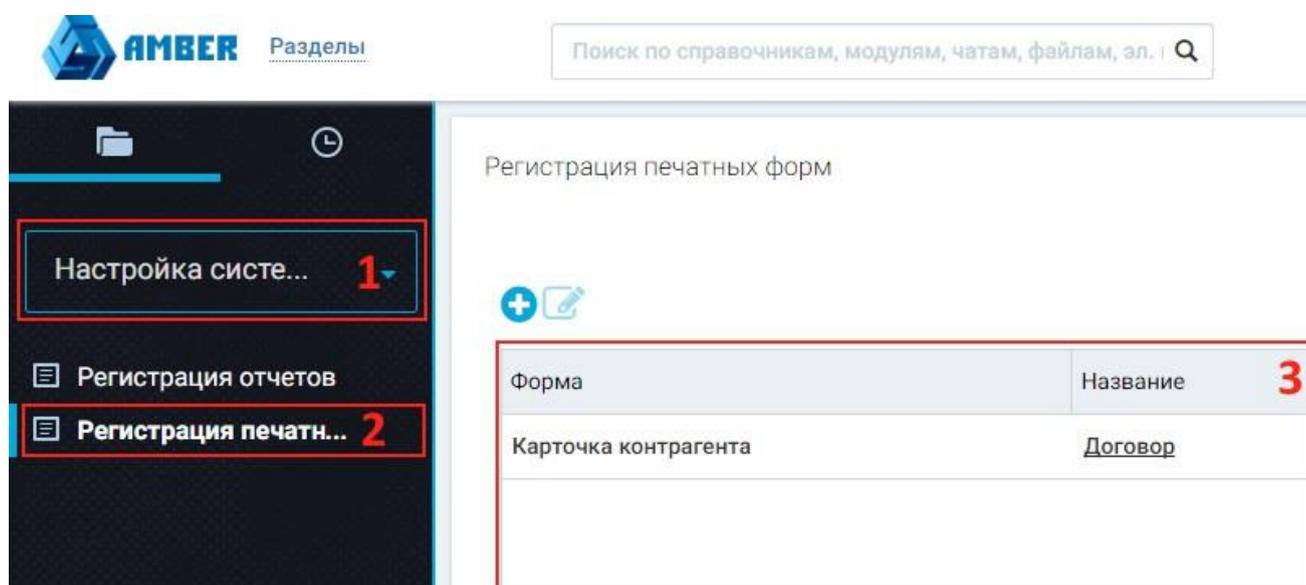


Рисунок 16. Регистрация печатных форм

- В списке рабочих мест (1 на рисунке) выбираем «Настройка системы»;
- Выбираем раздел «Регистрация печатных форм» (2 на рисунке);
- Для регистрации новой печатной формы нажимаем + (3 на рисунке);
- Откроется карточка регистрации новой печатной формы и табличных частей печатной формы;

Регистрация печатных форм • Карточка печатной формы

Название	Название представления	Форма
<input type="text" value="Договор"/>	<input type="text" value="V_PR_Contract"/>	<input type="text" value="Карточка контрагента"/>
Идентификатор печатной формы		
<input type="text" value="0"/>		
Шаблон печатной формы		
<input type="text" value="Лицензионное соглашение AMBER.d..."/>		

Метка	Название представления
<u>Договор.Продукты</u>	V_PR_TB_ProductsInContract

Рисунок 17. Карточка регистрации печатной формы

Описание полей карточки печатной формы:

- **Название** – отображаемое в системе печатной формы;
- **Название представления** – указываем ранее созданное представление;
- **Форма** – выбираем форму в которой будет отображаться текущая печатная форма;
- **Идентификатор печатной формы** – идентификатор печатной формы в системе. Если поле идентификатор печатной формы заполнено, то происходит обновление данных печатной формы. Если поле идентификатор печатной формы пустое, то происходит регистрация новой печатной формы;
- **Шаблон печатной формы** – место для загрузки файла печатной формы. Можно либо перетянуть файл на эту область, либо нажать и выбрать файл из обозревателя файлов;
- **Деталь табличные метки** - заводим список табличных меток документа.

Регистрация печатных форм • Карточка печатной формы • Карточка табличная часть печатной формы

Метка	Название представления	Идентификатор табличной части
<input type="text" value="Договор.Продукты"/>	<input type="text" value="V_PR_TB_ProductsInContract"/>	<input type="text" value="0"/>

Рисунок 18. Карточка метки

Описание полей карточки метки:

- **Метка** – название табличной метки в файле word;
- **Название представления** – указываем ранее созданное представление для табличной части;

- **Идентификатор табличной части** – идентификатор табличной части в системе. Если поле идентификатор табличной части заполнено, то происходит обновление данных табличной части. Если поле идентификатор табличной части пустое, то происходит регистрация новой табличной части;

6. Заполняем данные печатных форм/табличных частей печатных форм и после завершения настройки печатных форм нажимаем «Сохранить».

5.2.3. Редактирование печатной формы в системе

Сценарий редактирования печатной формы Amber практически совпадает со сценарием добавления новых печатных форм. Отличие в том, что печатная форма уже существует в системе

1. Заходим в систему;

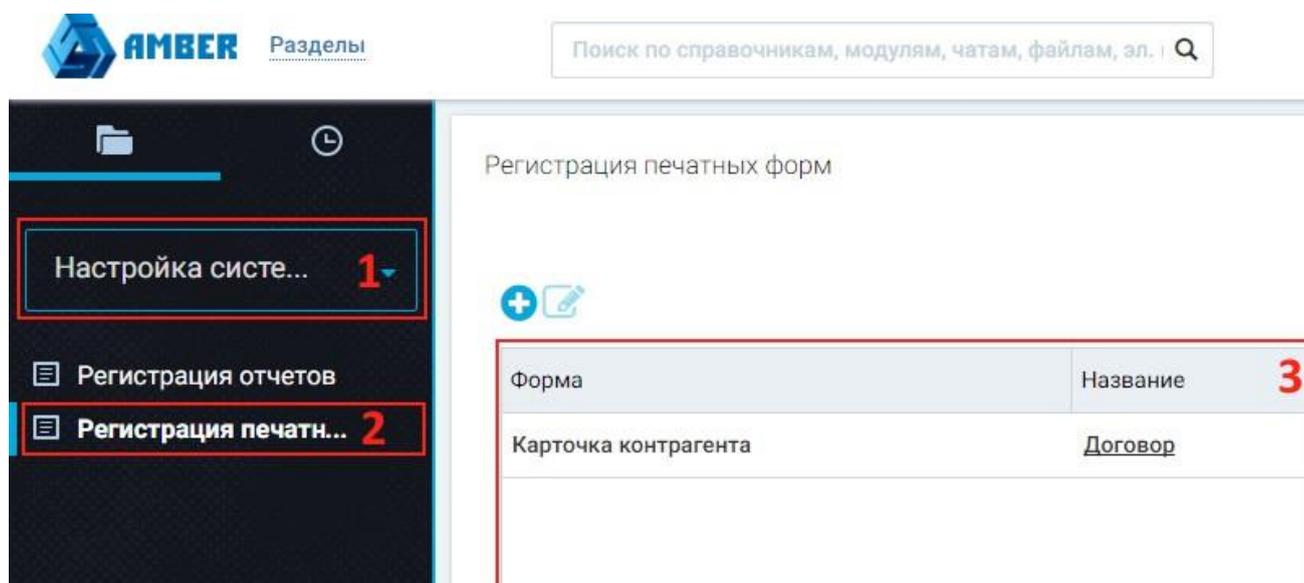


Рисунок 19. Регистрация печатных форм

2. В списке рабочих мест (1 на рисунке) выбираем «Настройка системы»;
3. Выбираем раздел «Регистрация печатных форм» (2 на рисунке);
4. Открываем существующую печатную форму для последующего редактирования (3 на рисунке);
5. Откроется карточка печатной формы;

Регистрация печатных форм - Карточка печатной формы

Название <input type="text" value="Договор"/>	Название представления <input type="text" value="V_PR_Contract"/>	Форма <input type="text" value="Карточка контрагента"/>
Идентификатор печатной формы <input type="text" value="0"/>		
Шаблон печатной формы <div style="border: 1px dashed blue; padding: 2px; display: inline-block;"> Лицензионное соглашение AMBER.d... ✕ </div>		




Метка	Название представления
Договор.Продукты	V_PR_TB_ProductsInContract

Рисунок 20. Карточка регистрации печатной формы

Описание полей карточки печатной формы:

- **Название** – отображаемое в системе печатной формы;
- **Название представления** – указываем ранее созданное представление;
- **Форма** – выбираем форму в которой будет отображаться текущая печатная форма;
- **Идентификатор печатной формы** – идентификатор печатной формы в системе. Если поле идентификатор печатной формы заполнено, то происходит обновление данных печатной формы. Если поле идентификатор печатной формы пустое, то происходит регистрация новой печатной формы;
- **Шаблон печатной формы** – место для загрузки файла печатной формы. Можно либо перетянуть файл на эту область, либо нажать и выбрать файл из обозревателя файлов;
- **Деталь табличные метки** - заводим список табличных меток документа.

б. Заполняем данные печатных форм/табличных частей печатных форм и после завершения настройки печатных форм нажимаем «Сохранить».

5.2.4. Удаление существующей печатной формы

Для удаления печатной формы производим следующие шаги:

1. Заходим в систему;

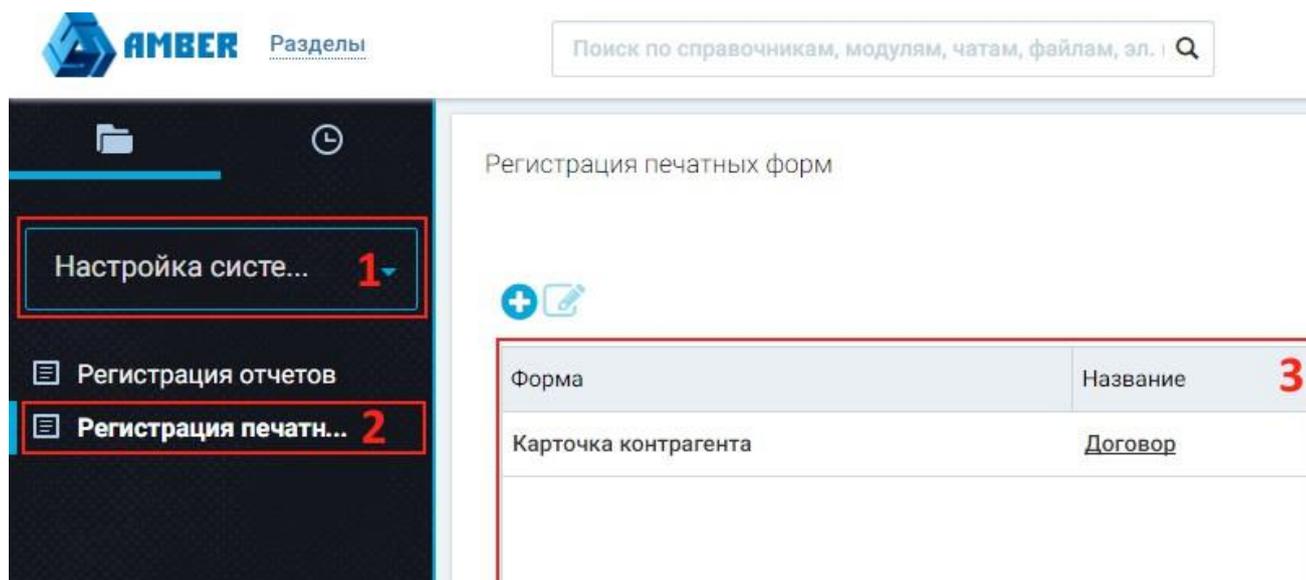


Рисунок 21. Регистрация печатных форм

- В списке рабочих мест (1 на рисунке) выбираем «Настройка системы»;
- Выбираем раздел «Регистрация печатных форм» (2 на рисунке);
- Удаляем существующую печатную форму.

5.2.5. Пример использования

- Создаем новое представление в БД

Основное представление

```
CREATE VIEW [BusinessBase].[V_PR_KP]
AS
SELECT o.Id, IsNull(gendir.FIO, '') AS 'КП.Гендир',
IsNull(client.LegalName, '') AS 'КП.ЮрНазвание'
FROM BusinessBase.Opportunity o
LEFT OUTER JOIN BusinessBase.BasePartner AS client ON client.Id = o.[Partner]
left outer join BusinessBase.BaseContact gendir ON gendir.Id = client.LPR left
outer join BusinessBase.Job j ON j.Id = gendir.Job
```

Представление для табличной части

```
CREATE VIEW [BusinessBase].[V_PR_DealProducts]
AS
SELECT pid.Deal as RecordId, p.Name as [Продукт], pio.Amount as [Количество],
pio.SumWithTax as 'Стоимость с НДС'
FROM BusinessBase.ProductInDeal pid
LEFT OUTER JOIN BusinessBase.ProductInObject pio on pio.Id = pid.Id
LEFT OUTER JOIN BusinessBase.BaseProduct p on p.Id = pio.Product
```

- Зарегистрировать печатную форму и табличные части в системе;
- Заходим в систему;

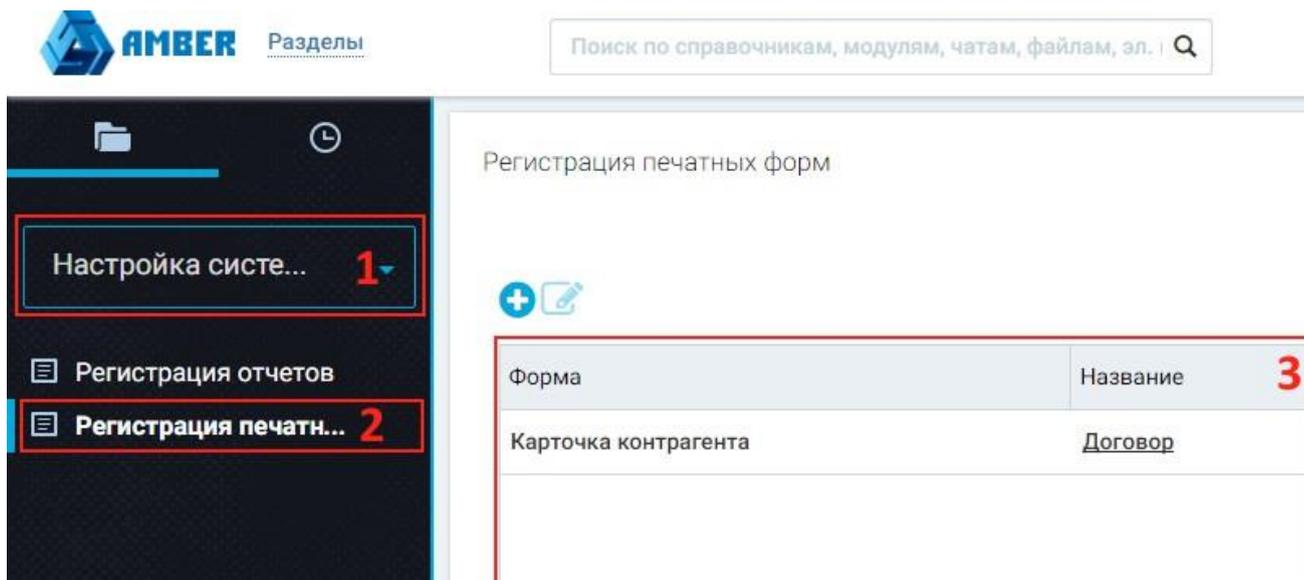


Рисунок 22. Регистрация печатных форм

4. В списке рабочих мест (1 на рисунке) выбираем «Настройка системы»;
5. Выбираем раздел «Регистрация печатных форм» (2 на рисунке);
6. Для регистрации новой печатной формы нажимаем + (3 на рисунке);
7. Откроется карточки регистрации новой печатной формы и табличных частей печатной формы;

Регистрация печатных форм - Карточка печатной формы

Название

Название представления

Форма

Идентификатор печатной формы

Шаблон печатной формы

[Лицензионное соглашение AMBER.d...](#) x

+ ✎

Метка	Название представления
Договор.Продукты	V_PR_TB_ProductsInContract

⚙

Рисунок 23. Карточка регистрации печатной формы

Описание полей карточки печатной формы:

- **Название** – отображаемое в системе печатной формы;
- **Название представления** – указываем ранее созданное представление;
- **Форма** – выбираем форму в которой будет отображаться текущая печатная форма;

- **Идентификатор печатной формы** – идентификатор печатной формы в системе. Если поле идентификатор печатной формы заполнено, то происходит обновление данных печатной формы. Если поле идентификатор печатной формы пустое, то происходит регистрация новой печатной формы;
 - **Шаблон печатной формы** – место для загрузки файла печатной формы. Можно либо перетянуть файл на эту область, либо нажать и выбрать файл из обозревателя файлов;
 - **Деталь табличные метки** - заводим список табличных меток документа.
8. Заполняем поля печатной формы и табличных частей. Файл с печатной формой пока не подтягиваем;
 9. Создаем новый word файл с метками. Названия текстовых меток должны соответствовать названию полей в представлении. Название табличных меток в шаблоне должны соответствовать заведенным при регистрации печатной формы табличным меткам;
 10. Загружаем файл печатной формы в ранее зарегистрированную печатную форму и нажимаем «Сохранить».

6. Описание API системы

6.1. Описание свойств объектов структуры

Свойства и значения справочников, помеченные курсивом и серым цветом, не являются значимыми и могут не приниматься во внимание при чтении модели структуры объекта.

6.1.1. Объект Structure

Code	Уникальный код объекта в системе.
BaseClassCode	Уникальный код объекта, от которого унаследован текущий объект. Если отсутствует, то объект не наследуется от кого-либо.
Caption	Массив объектов Localization с локализациями. Содержит краткое название объекта.
Description	Массив объектов Localization с локализациями. Содержит подробное описание объекта.
CreateDate	Дата создания объекта
CreatorUserId	Идентификатор пользователя, создавшего объект
UpdatedDate	Дата последнего обновления объекта.
UpdaterUserId	Идентификатор пользователя, последним изменившего объект.
CustomObjectTypeCode	Уникальный код типа объекта.
DefaultViewProperty	Уникальный код свойства, используемого как текстовое значение при замене идентификатора текстом или в случае выбора данных в словарь ключ-значение, где ключом является идентификатор объекта, а значение берется из свойства с кодом, указанным в DefaultViewProperty
Properties	Массив объектов Property. Содержит описание свойств объекта.
<i>Events</i>	События объекта.
DependencyCodes	Массив уникальных кодов объектов, которые ссылаются на данный объект

6.1.2. Объект Property

Caption	Массив объектов Localization с локализациями. Содержит краткое название свойства.
Code	Уникальный код свойства в системе.
ClassCode	Уникальный код объекта, в котором объявлено данное свойство.
<i>IsOverridable</i>	Можно ли переопределять свойство в потомках

IsEditable	Возможно ли изменять структуру свойства
IsReadOnly	Можно ли создавать, изменять, удалять данные этого свойства
IsSystem	Свойство является системным.
IsVisible	Видимо ли свойство.
ClassReference	Если значение свойства не пустое, значит это свойство является ссылкой один-ко-многим и содержит описание объекта, на который ссылается.
Numerator	Содержит объект Numerator с настройками для данного свойства.
PropertyType	Идентификатор типа свойства.
Stored	Если значение свойства не пустое, значит это свойство является хранимым.
SyntheticProperty	Информация о типе данных свойства и правилах, применяемых к свойству. Например ограничения по длине строки или значение по умолчанию.

6.1.3. Значения справочника Property types:

Идентификатор	Код типа	
0	Stored Key	Свойство является идентификатором строк данных объекта. Хранимое в базе данных свойство.
1	Run-Time	
2	Stored	Хранимое в базе данных свойство.
4	Class Ref.	Свойство, представляющее из себя ссылку один-ко-многим на другой объект и содержит идентификатор записи, на которую ссылается объект.
7	Numerator	Свойство является нумератором
8	Key	Свойство является идентификатором строк данных объекта.
9	ClassProperty	

6.1.4. Объект ClassReference

CascadeDeleteTypeCode	Действие при каскадном удалении.
ClassCode	Уникальный код объекта, на который ссылается свойство.
ResultPropertyCode	Уникальный код свойства, используемого для отображения текстовой информации по имеющемуся идентификатору.

6.1.5. Объект Localization

Code	Код локали. Например ru или en.
------	---------------------------------

Value	Локализованное строковое значение.
-------	------------------------------------

6.1.6. Объект SyntheticProperty

SyntheticTypeCode	Уникальный код типа данных
RuleValues	Массив правил для свойства

6.1.7. Объект RuleValue

<i>CalcTypeCode</i>	
Code	Уникальный код правила
Value	Значение для правила

6.1.8. Объект Numerator

Arguments	Массив объектов NumeratorArgument
-----------	-----------------------------------

6.1.9. Объект NumeratorArgument

Type	Тип нумератора. Generator – автогенерируемое значение Constant – константное значение
GeneratorArg	Настройки автогенерируемого значения.
ConstantArg	Настройки константного значения.

6.2. Аутентификация

Процесс аутентификации осуществляется путем запроса токена по логину/паролю.

Данный токен должен использоваться в каждом запросе к API в виде HTTP заголовка с именем "token" и значением, полученным в ответе на запрос аутентификации.

Запрос:

На сервер отсылается JSON объект с определенным составом полей.

HTTP URL:

POST `http://<amber_host>/API/V1.svc/auth`

HTTP URL параметры:

Не требуются.

HTTP заголовки запроса:

Content-Type: application/json

HTTP тело запроса:

```
{
  "Login":<user_login>,
  "Password":< user_password>,
  "AppId": "1"
}
```

Ответ:

JSON объект с токеном аутентификации.

HTTP статусы ответа:

200 – токен получен.

HTTP тело ответа:

```
{
  "Token": "6e49bfdb804c4aaaa2073e35524fba27"
}
```

6.3. Получение информации по объекту

Позволяет получить детальную информацию о структуре объекта по уникальному коду.

Запрос:

HTTP URL:

GET `http:// <amber_host>/API/V1.svc/structure/{customObjectCode}`

HTTP URL параметры:

customObjectCode	обязательный, строка. Уникальный идентификатор объекта в рамках системы.
------------------	--

HTTP заголовки запроса:

token: <token>

HTTP тело запроса:

Не требуется.

Ответ:

JSON объект с детальным описанием структуры объекта.

HTTP статусы ответа:

200 – объект найден.

404 – объект не найден.

HTTP тело ответа:

Описание структуры содержится в разделе 2.

Пример:

```
{  
  "BaseClassCode" : "BaseCodeDictionary",  
  "Caption" : [{ "Code" : "ru", "Value" : "Специализация" }],  
  "Code" : "Specialization",  
  "CreateDate" : {  
    "DateTime" : "/Date(1457966749158)/",
```

```
        "OffsetMinutes" : 0
    },
    "CreatorUserId" : null,
    "CustomObjectTypeCode" : "BusinessObjectPreset",
    "DefaultViewProperty" : "Name",
    "DependencyCodes" : [
        "BasePartner",
        "MarketPart",
        "BaseLead"
    ],
    "Description" : [{ "Code" : "ru", "Value" : "" }],
    "Events" : [],
    "GroupOfDuplicateRules" : [],
    "Properties" : [{
        "Caption" : [{ "Code" : "ru", "Value" : "Отрасль"}],
        "ClassCode" : "Specialization",
        "ClassReference" : {
            "CascadeDeleteTypeCode" : "SETNULL",
            "ClassCode" : "Branch",
            "ResultPropertyCode" : "Name"
        },
        "Code" : "Branch",
        "IsEditable" : false,
        "IsOverridable" : false,
        "IsReadOnly" : false,
        "IsSystem" : false,
        "IsVisible" : true,
        "Numerator" : null,
        "PropertyType" : 2,
        "Stored" : {
            "ColumnName" : "Branch",
            "IsAllowNull" : true
        }
    }]
```

```
} {  
    "SyntheticProperty" : null
```

```
"Caption" : [{ "Code" : "ru", "Value" : "Название" }],
"ClassCode" : "BaseCodeDictionary", "ClassReference"
: null,
"Code" : "Name",
"IsEditable" : false,
"IsOverridable" : false,
"IsReadOnly" : false,
"IsSystem" : false,
"IsVisible" : true,
"Numerator" : null,
"PropertyType" : 3,
"Stored" : {
    "ColumnName" : "Name", "IsAllowNull"
    : false
},
"SyntheticProperty" : {
    "RuleValues" : [{
        "CalcTypeCode" : "ConstantValue", "Code"
        : "MaxLength",
        "Value" : "200"
    }, {
        "CalcTypeCode" : "ConstantValue", "Code"
        : "MinLength",
        "Value" : "0"
    }
    ],
    "SyntheticTypeCode" : "string"
}
},
"SearchProperties" : [
    "Name"
],
```

```
"UpdatedDate" : {  
    "DateTime" : "/Date(1457966749158)/",
```

```
        "OffsetMinutes" : 0
    },
    "UpdaterUserId" : null
}
```

6.4. Создание записи данных объекта

Позволяет создать записи данных для объекта с учетом прав пользователя. Для связывания справочников один-ко-многим используются идентификаторы.

Запрос:

На сервер отсылается JSON объект с динамическим составом полей.

Каждое поле объекта формируется согласно структуре в виде “Code свойства”: <значение>.

HTTP URL:

POST [http:// <amber_host>/API/V1.svc/instance/{customObjectCode}](http://<amber_host>/API/V1.svc/instance/{customObjectCode})

HTTP URL параметры:

customObjectCode	обязательный, строка. Уникальный идентификатор объекта в рамках системы.
------------------	--

HTTP заголовки запроса:

token: <token>

HTTP тело запроса:

```
{
  "Code": "testApi1",
  "Name": "Неконкурентная цена",
  "Description": "Some description",
  "Branch": <id записи из справочника Branch>
}
```

Ответ:

JSON объект с идентификатором созданной записи.

HTTP статусы ответа:

200 – объект создан.

HTTP тело ответа:

```
{  
  "InstanceId": 458  
}
```

6.5. Редактирование записи данных объекта

Позволяет редактировать записи данных для объекта с учетом прав пользователя.

Запрос:

На сервер отсылается JSON объект с динамическим составом редактируемых полей.

Каждое поле объекта формируется согласно структуре в виде “Code свойства”: <значение>.

Поля, не нуждающиеся в редактировании, в объект не включаются.

HTTP URL:

PUT [http:// <amber_host>/API/V1.svc/instance/{customObjectCode}/{id}](http://<amber_host>/API/V1.svc/instance/{customObjectCode}/{id})

HTTP URL параметры:

customObjectCode	обязательный, строка. Уникальный идентификатор объекта в рамках системы.
id	обязательный, целочисленный. Идентификатор записи данных.

HTTP заголовки запроса:

token: <token>

HTTP тело запроса:

```
{  
  "Description": "new description"  
}
```

Ответ:

HTTP статусы ответа:

200 – объект отредактирован.

6.6. Удаление записи данных объекта

Позволяет удалять записи данных для объекта с учетом прав пользователя.

Запрос:

HTTP URL:

DELETE [http:// <amber_host>/API/V1.svc/instance/{customObjectCode}/{id}](http://<amber_host>/API/V1.svc/instance/{customObjectCode}/{id})

HTTP URL параметры:

customObjectCode	обязательный, строка. Уникальный идентификатор объекта в рамках системы.
Id	обязательный, целочисленный. Идентификатор записи данных.

HTTP заголовки запроса:

token: <token>

HTTP тело запроса:

Не требуется.

Ответ:

HTTP статусы ответа:

200 – объект удален.

6.7. Получение списка записей данных объекта с возможностью фильтрации и пейджинга

Запрос:

HTTP URL:

[http://<amber_host>/API/V1.svc/instances/{customObjectCode}?size={size}&page={page}&conditions=\[{"Left":"Code","Right":"testApi1","Operator":"Equal"}\]](http://<amber_host>/API/V1.svc/instances/{customObjectCode}?size={size}&page={page}&conditions=[{)

HTTP URL параметры:

Название:	Обязательность	Тип	По умолчанию	
customObjectCode	да	строка	нет	Уникальный идентификатор объекта в рамках системы.
size	нет	целочисленный	50	Размер записей данных, включенный в выбираемую страницу данных. Сейчас полностью игнорируется и размер всегда равен 50.
page	нет		1	Номер выбираемой страницы данных. Начинается с 1.
conditions	нет	Массив условий в виде JSON объектов	нет	<p>Соединение условий происходит по OR.</p> <p>Пример:</p> <pre>{ "Left" : "Code", "Right" : "testApi1", "Operator" : "Equal" }, { "Left" : "Code", "Right" : "testApi2", "Operator" : "Equal" }</pre>

Описание JSON объекта условия фильтрации:

- Left – Code свойства объекта
- Right – значение фильтрации для свойства
- Operator – оператор.

Список поддерживаемых операторов: Equal, NotEqual, LessThan, LessThanOrEqual, GreatThan, GreatThanOrEqual, IsNull, NotNull, StartsWith, Contains, EndsWith

HTTP заголовки запроса:

token: <token>

HTTP тело запроса:

Не требуется.

Ответ:

HTTP статусы ответа:

200.

HTTP тело ответа:

Массив JSON объектов данных.

Пример:

```
[
  {
    "Id": "7246",
    "Branch": "28",
    "Name": "Неконкурентная цена",
    "Code": "testApi1",
    "Description": "aaabbbccc",
    "CustObjVersion": "823",
    "CreateDate": "2016-07-12T12:21:23.0325906+00:00",
    "UpdateDate": "2016-07-12T12:21:23.0325906+00:00"
  },
  {
    "Id": "7247",
    "Branch": "28",
    "Name": "Неконкурентная цена",
    "Code": "testApi2",
    "Description": "aaabbbccc",
    "CustObjVersion": "823",
    "CreateDate": "2016-07-12T12:24:36.3975906+00:00",
```

```
"updateDate": "2016-07-12T12:24:36.3975906+00:00"  
}  
]
```

6.8. Получение записи по идентификатору

Запрос:

HTTP URL:

http:// <amber_host>/API/V1.svc/instance/{customObjectCode}/{id}

HTTP URL параметры:

customObjectCode	обязательный, строка. Уникальный идентификатор объекта в рамках системы.
id	обязательный, целочисленный. Идентификатор записи данных.

HTTP заголовки запроса:

token: <token>

HTTP тело запроса:

Не требуется.

Ответ:

HTTP статусы ответа:

200.

HTTP тело ответа:

JSON объект данных.

Пример:

```
{  
  "Id": "7247",  
  "Branch": "28",  
  "Name": "Неконкурентная цена",  
  "Code": "testApi2",  
  "Description": "aaabbbccc",  
  "CustObjVersion": "823",  
}
```

```
"CreateDate": "2016-07-12T12:24:36.3975906+00:00",  
"UpdateDate": "2016-07-12T12:24:36.3975906+00:00"  
}
```

6.9. Получение данных объекта по «расширенному» запросу

На сервер отсылается модель InstanceQuery в формате JSON.

Описание полей InstanceQuery:

Название	Тип	Описание
CustomObjectCode	Строка	Уникальный идентификатор объекта в рамках системы.
Select	PropertyPath[]	Загружаемые свойства объекта.
Where	ExpressionElement[]	Условие фильтрации данных.
OrderBy	OrderedPropertyPath[]	Сортировка данных.
Offset	Целое	Смещение выборки данных.
PageSize	Целое	Размер выборки данных. Максимальное значение – 50.

Описание полей PropertyPath:

Название	Тип	Описание
Property	Строка	Код свойства объекта. Поддерживается обращение к ссылочным полям через точку. Пример – Owner.State.Name, где: <ul style="list-style-type: none"> • Owner – код собственного свойства загружаемого объекта (свойство ссылается на объект Contact); • S t • Name – код свойства объекта ContactState.

Описание полей OrderedPropertyPath:

Название	Тип	Описание
Property	Строка	Код свойства объекта. Поддерживается обращение к ссылочным полям через точку. Пример:
Order	Строка	Направление сортировки. Возможные значения: <ul style="list-style-type: none"> • A • sDesc – сортировка по убыванию.

Описание полей ExpressionElement:

Название	Тип	Описание
Brackets	Brackets	Скобки.
Constant	Constant	Константа.
Function	Function	Функция.

Operator	Operator	Оператор.
----------	----------	-----------

PropertyPath	PropertyPath	Свойство объекта.
--------------	--------------	-------------------

Описание полей Operator:

Название	Тип	Описание
OperatorCode	Строка	<p>Код оператора.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • EqualTo – равно; • GreaterThan – больше; • GreaterThanOrEqualTo – больше или равно; • LessThan – меньше; • LessThanOrEqualTo – меньше или равно; • NotEqualTo – не равно; • Plus – плюс; • Minus – минус; • Multiply – умножить; • Divide – разделить; • Modulo – остаток от деления; • And – логическое И; • Or – логическое ИЛИ; • IsNull – не заполнено. Формат: <Операнд> IsNull; • IsNotNull – заполнено. Формат: <Операнд> IsNotNull; • Not – логическое отрицание. Формат: Not <Операнд>; • MinusUnary – унарный минус. Формат: MinusUnary <Операнд>; • StartsWith – строка начинается со значения; • E • Ъ • OnDate – на дату. sn

Описание полей Function:

Название	Тип	Описание
FunctionCode	Строка	<p>Код функции.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • IIF – условие. Функция проверяет условие, записанное в первом аргументе. Если условие выполняется - возвращает значение второго аргумента. Если не выполняется - значение третьего

		аргумента; – текущее системное время UTC.
Arguments	E	Аргументы функции.

Описание полей Constant:

Название	Тип	Описание
DataTypeCode	Строка	Код типа данных *.
Value	Строка	Значение константы.

* нужный тип данных свойства можно получить из запроса метаданных объекта. Таблица соответствия:

SyntheticTypeCode	DataTypeCode
integer	Int
logical	Bit
string	String
text	LongString
identifier	UniqueIdentifier
decimal	Decimal
money	Money
datetime	LongDateTime
date	LongDateTime
time	Time
duration	BigInt
url	LongString
telephone	String
email	String
percent	Decimal
fileinfo	FileInfo
imageinfo	ImageInfo

Описание полей Brackets:

Название	Тип	Описание
Items	ExpressionElement[]	Элементы в скобках.

Запрос:

h

HTTP URL параметры:

Не требуются.

HTTP заголовки запроса:

Content-Type: application/json

HTTP тело запроса:

```
{
  "CustomObjectCode": "Contact",
  "Select": [
    { "Property": "Id" },
    { "Property": "FIO" },
    { "Property": "ContactState.Name" }
  ],
  "Where": [
    {
      "Brackets": {
        "Items": [
          { "PropertyPath": { "Property": "ContactType.Name" } },
          { "Operator": { "OperatorCode": "EqualTo" } },
          { "Constant": { "DataTypeCode": "String", "Value": "Наш сотрудник" } }
        ]
      }
    },
    { "Operator": { "OperatorCode": "And" } },
    {
      "Brackets": {
        "Items": [
          { "PropertyPath": { "Property": "FIO" } },
          { "Operator": { "OperatorCode": "StartsWith" } },
          { "Constant": { "DataTypeCode": "String", "Value": "User" } }
        ]
      }
    }
  ],
  "OrderBy": [
    {
      "Property": "FIO",
      "Order": "Asc"
    }
  ]
}
```

Ответ:

HTTP статусы ответа:

HTTP тело ответа:

Массив JSON объектов данных.

Пример:

```
[  
  {  
    "Id": "9909",  
    "FIO": "User9909",  
    "ContactState.Name": "Активный"  
  },  
  {  
    "Id": "10062",  
    "FIO": "User10062",  
    "ContactState.Name": "Активный"  
  }  
]
```