

## Содержание

1) Сформировать в ADVANTA LINQ-запросы .....	3
Пример LINQ-запроса данных по проектам .....	3
Пример LINQ-запроса данных по контрольным точкам .....	4
2) Создать шаблоны таблиц данных в формате .xlsx .....	5
Пример шаблона для источника данных по проектам .....	6
Пример шаблона для источника данных по контрольным точкам .....	6
3) Создать модель в Analytic Workspace и сформировать скрипт в ETL-редакторе .....	7
Пример скрипта для проектов .....	8
Пример скрипта для контрольных точек .....	10
Пример модели .....	12
4) Создать виджеты для отображения данных и дашборды из виджетов .....	12
Пример дашборда «Здоровье портфеля» в Analytic Workspace на основе данных, полученных из ADVANTA .....	12



# Интеграция ADVANTA с Analytic Workspace

[Analytic Workspace](#) — российская BI-система, позволяющая обрабатывать данные из разных источников и визуализировать их при помощи широкого набора встроенных инструментов. Благодаря интеграции возможно использование инструментария Analytic Workspace для построения дашбордов на основе данных из ADVANTA. Это дает дополнительные возможности для анализа данных и последующего формирования управленческих решений.

Ниже приведена подробная пошаговая инструкция по организации процесса обмена данными между двумя системами. Для реализации обмена у вас должен быть доступ к по крайней мере к демо-версиям данных продуктов.

## 1) Сформировать в ADVANTA LINQ-запросы

Интеграция с Analytic Workspace организована с использованием [LINQ-запросов](#) и [Web-API](#) ADVANTA. Таким образом, для того, чтобы загрузить данные в BI-систему, необходимо предварительно сформировать LINQ-запросы внутри ADVANTA для получения тех данных, которые в дальнейшем будут визуализированы.

### Пример LINQ-запроса данных по проектам

```
/* 1. перечень всех проектов (2 типа объектов - IT-проект и Организационный проект)
с UID и всеми системными реквизитами (сроки начала/завершения - план, факт,
ответственные (ФИО), статус (текстом), последний базовый план)
- объект-родитель (название)
и пользовательскими реквизитами:
- жизненный цикл проекта
- описание
- масштаб проекта (текстовое значение)
- эффект
- последняя запись из справочника "Отчет о статусе" - все поля
- активная запись из справочника "Бюджет проекта" - только поле "Бюджет (план)"
*/
```

```
var projects = dataContext.Projects
    .Where(p => p is IT_proekt_d46b6d || p is
Organizacionnij_proekt_a3e2fb)
    .OrderBy(p => p.CreationDate)
    .Select(p => new { p.Id,
                    p.Name, ParentName = p.Parent.Name,
                    LifeCycleName =
p.Fields.Zhiznennij_cikl_proekta_b07f57.Name,
                    p.SystemStartDate, p.SystemEndDate,
p.ActualStartDate, p.ActualEndDate,
                    p.BaselinePlanStartDate, p.BaselinePlanEndDate,
                    BossFIO = String.Concat( p.Owner.LastName, " ",
p.Owner.FirstName), RPFIO = String.Concat( p.Responsible.LastName, " ",
```

```

p.Responsible.FirstName),
        Description = p.Fields.Opisanie_caaaaa,
        ProjectScaleName =
p.Fields.Masshtab_proekta_04412f.Name,
        ProjectGainName = p.Fields.Effekt_b6c145.Name,
        StatusReport_Date =
p.GetChildren<Otchet_o_stavke_440616>().OrderByDescending(o =>
o.Date).FirstOrDefault().Date,
        StatusReport_Chto_sdelano_problemi_riski =
p.GetChildren<Otchet_o_stavke_440616>().OrderByDescending(o =>
o.Date).FirstOrDefault().Chto_sdelano_problemi_riski_4a11b8,
        StatusReport_Prichina_problemi =
p.GetChildren<Otchet_o_stavke_440616>().OrderByDescending(o =>
o.Date).FirstOrDefault().Prichina_problemi_a3e564.Name,
        StatusReport_StatusName =
p.GetChildren<Otchet_o_stavke_440616>().OrderByDescending(o =>
o.Date).FirstOrDefault().Status_e05ba3.Name,
        BudjetSum =
p.GetChildren<Byudzheto_plan_e5bee7>().Where(b =>
b.Aktivnostj_versii_281a38.Id ==
Classifier_Aktivnostj_versii_625508.Aktivnaya_4f5ba775_Id).Sum(b =>
b.Summa_06ff93)
    });
return projects;

```

## Пример LINQ-запроса данных по контрольным точкам

```

/*
2. перечень всех контрольных точек (1 тип объекта - КТО)
с UID и всеми системными реквизитами (как и у проекта),
- UID - проекта-родителя (непрямой родитель)
и пользовательские поля:
- описание (текст)
- результат (текст)
- последняя запись из справочника "Отчет о мероприятии" - все поля
*/
var projects = dataContext.KT0_1ff431_List
    .Select(p => new {
        p.Id,
        p.Name,
        ProjectId =
(Guid?)p.GetParentHierarchy<Project>(false).Where(p => p is IT_proekt_d46b6d
|| p is Organizacionnij_proekt_a3e2fb).FirstOrDefault().Id,
        // ProjectName =
p.GetParentHierarchy<Project>(false).Where(p => p is IT_proekt_d46b6d || p
is Organizacionnij_proekt_a3e2fb).FirstOrDefault().Name,
        p.SystemStartDate, p.SystemEndDate, p.ActualStartDate,

```

```
p.ActualEndDate,
    p.BaselinePlanStartDate, p.BaselinePlanEndDate,
    BossFIO = String.Concat( p.Owner.LastName, " ",
p.Owner.FirstName), RPFIO = String.Concat( p.Responsible.LastName, " ",
p.Responsible.FirstName),
    Description = p.Opisanie_caaaaa,
    Result = p.Rezuljtat_8e716f,
    StatusReport_Date =
p.GetChildren<Otchet_o_stature_meropriyatiya_2bc942>().OrderByDescending(o
=> o.Date).FirstOrDefault().Date,
    StatusReport_Chto_sdelano_problemi_riski =
p.GetChildren<Otchet_o_stature_meropriyatiya_2bc942>().OrderByDescending(o
=> o.Date).FirstOrDefault().Chto_sdelano_problemi_riski_4a11b8,
    StatusReport_Prichina_problemi =
p.GetChildren<Otchet_o_stature_meropriyatiya_2bc942>().OrderByDescending(o
=> o.Date).FirstOrDefault().Prichina_problemi_a3e564.Name,
    StatusReport_StatusName =
p.GetChildren<Otchet_o_stature_meropriyatiya_2bc942>().OrderByDescending(o
=> o.Date).FirstOrDefault().Status_e05ba3.Name
    });
return projects;
```

Данный код запросов приведен в качестве примера. На вашей инсталляции необходимо будет указать наименования сущностей, объектов, справочников, реквизитов, соответствующих сформированному контексту LINQ-запросов.

## 2) Создать шаблоны таблиц данных в формате .xlsx

Данные шаблоны необходимы для формирования структуры хранения данных в AW в соответствии со структурой получаемых их ADVANTA данных.

Для создания шаблона необходимо **выполнить созданный LINQ-запрос** в ADVANTA, скопировать шапку полученной таблицы в Excel и сохранить файл в формате .xlsx.

The screenshot shows the ADVANTA interface with a query editor and a data table. The query editor contains a complex SQL query with C#-style syntax. Below the query editor, there are buttons for 'Сделать запрос' (Make query), 'Сохранить' (Save), and 'Удалить' (Delete). The 'Сделать запрос' button is highlighted with a red box. Below the buttons, there is a checkbox for 'Может использоваться в диаграмме Ганта' and another for 'Отображать SQL-запрос'. The data table below shows columns for project details and budget information.

Id	Name	ParentName	LifeCycleName	SystemStartDate	SystemEndDate	ActualStartDate	ActualEndDate	BaselinePlanStart	BaselinePlanEnd	BossFIO	RFFIO	Description	ProjectScaleN	ProjectGainN	StatusReport_1	StatusReport_2	StatusReport_3	StatusReport_4	BudgetSum
												Настройка							

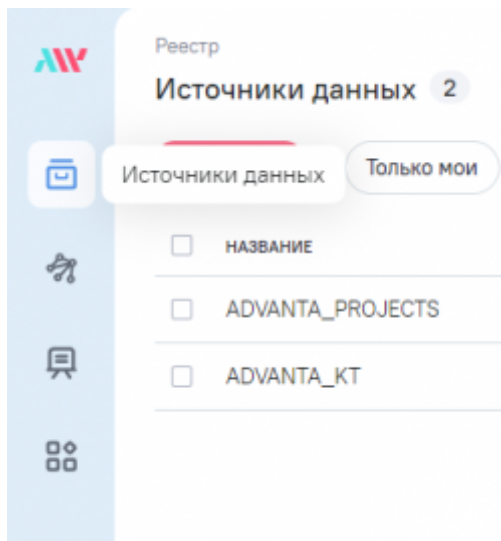
### Пример шаблона для источника данных по проектам

	A	B	C	D	E	F	G	H
1	Id	Name	ParentName	LifeCycleName	SystemStartDate	SystemEndDate	ActualStartDate	ActualEndDate
2								
3								
4								
5								
6								

### Пример шаблона для источника данных по контрольным точкам

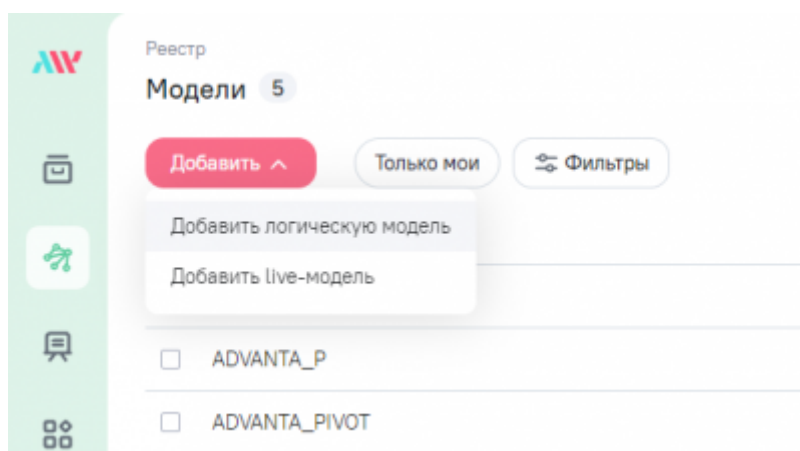
	A	B	C	D	E	F	G	H
1	Id	Name	ProjectId	SystemStartDate	SystemEndDate	ActualStartDate	ActualEndDate	BaselinePlanStartDate
2								
3								
4								
5								

Загрузите данные шаблоны в Analytic Workspace, в раздел «Источники данных» — здесь необходимо **создать источник, выбрать формат «Файл» и загрузить сформированный шаблон.**

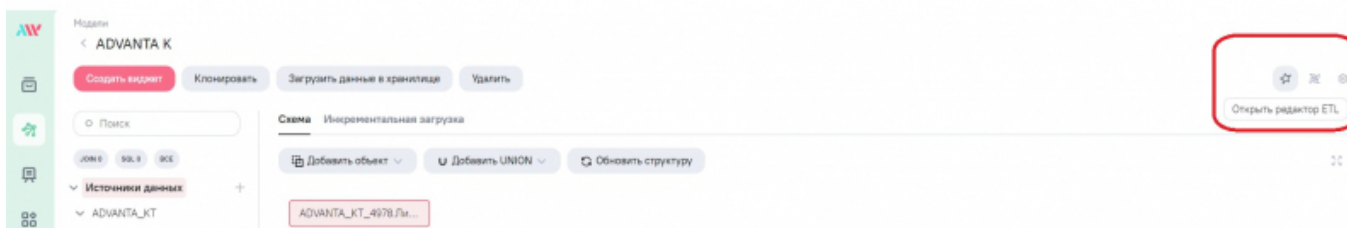


### 3) Создать модель в Analytic Workspace и сформировать скрипт в ETL-редакторе

После создания источника необходимо в соответствующем разделе Analytic Workspace **создать модель**, которая и будет служить витриной данных для созданного источника. Необходимо создать столько же моделей, сколько предполагается использовать LINQ-запросов.



Сразу после создания необходимо перейти в режим редактирования модели и зайти в **ETL-редактор**.



В открывшемся редакторе необходимо ввести скрипт для обработки получаемых из ADVANTA данных. Скрипт написан на Python.

## Пример скрипта для проектов

```
# -----  
-----  
# Скрипт для обработки модели ***  
# -----  
-----  
  
import requests  
import datetime  
from pyspark.sql import Row  
  
# python - m pip install requests, matplotlib, pandas  
  
from pandas import json_normalize  
  
def after_all(df, spark, app, *args, **kwargs):  
    print(df.schema)  
  
    LOGIN = '* * *'  
    PASSWORD = '* * *'  
    DOMAIN = 'https://* * *.ru'  
  
    AUTH = {  
        'Login': LOGIN,  
        'Password': PASSWORD,  
    }  
  
    session = requests.Session()  
    # cookies = session.cookies.get_dict() # {}  
  
    # авторизация  
    response = session.post(url=DOMAIN+'/api/auth/login', json=AUTH,  
verify=False)  
  
    cookies = session.cookies.get_dict()  
  
    # print(cookies)  
  
    # получение LINQ-запроса  
    LINQ = {  
        'DataSourceKey': 'BI_projects',  
        # нужен другой запрос для перечня проектов в модель list1  
        # 'DataSourceKey': 'BI_projects',  
        # и еще один для перечня КТ из этих проектов в модель list1_ywrq  
        # 'DataSourceKey': 'BI_milestones',  
        'PageSize': 100,  
    }  
}
```

```
r = session.post(url=DOMAIN+'/api/queries/get', cookies=cookies,
json=LINQ, verify=False)

if not r.ok:
    raise Exception(f'Ошибка в {r.url}. HTTP {r.status_code}: {r.text}')

data = r.json()

#Создадим список, который послужит основой для создания DataFrame
rows = []
for line in data:
    rows.append(Row(
        id=line['Id'],
        name=line['Name'],
        parentname=line['ParentName'],
        lifecyclename=line['LifeCycleName'],
        #
systemstartdate=str(datetime.datetime.fromisoformat(line['SystemStartDate'])
.replace(tzinfo=datetime.timezone.utc)),
        #
systemenddate=str(datetime.datetime.fromisoformat(line['SystemEndDate'])).rep
lace(tzinfo=datetime.timezone.utc)),
        #
actualstartdate=str(datetime.datetime.fromisoformat(line['ActualStartDate']))
.replace(tzinfo=datetime.timezone.utc)),
        #
actualenddate=str(datetime.datetime.fromisoformat(line['ActualEndDate'])).rep
lace(tzinfo=datetime.timezone.utc)),
        #
baselineplanstartdate=str(datetime.datetime.fromisoformat(line['BaselinePlan
StartDate'])).replace(tzinfo=datetime.timezone.utc)),
        #
baselineplanenddate=str(datetime.datetime.fromisoformat(line['BaselinePlanEn
dDate'])).replace(tzinfo=datetime.timezone.utc)),

        systemstartdate=line['SystemStartDate'],
        systemenddate=line['SystemEndDate'],
        actualstartdate=line['ActualStartDate'],
        actualenddate=line['ActualEndDate'],
        baselineplanstartdate=line['BaselinePlanStartDate'],
        baselineplanenddate=line['BaselinePlanEndDate'],

        bossfio=line['BossFIO'],
        rpfio=line['RPFIO'],
        description=line['Description'],
        projectscalename=line['ProjectScaleName'],
        projectgainname=line['ProjectGainName'],

        #
statusreport_date=str(datetime.datetime.fromisoformat(line['StatusReport_Dat
e'])).replace(tzinfo=datetime.timezone.utc)),
```

```

        statusreport_date=line['StatusReport_Date'],

statusreport_что_sделано_problemi_riski=line['StatusReport_Что_sделано_problemi_riski'],
statusreport_prichina_problemi=line['StatusReport_Prichina_problemi'],
        statusreport_statusname=line['StatusReport_StatusName'],
        budgetsum=line['BudgetSum']
    ))

return spark.createDataFrame(rows)

```

### Пример скрипта для контрольных точек

```

# # -----
-----
# Скрипт для обработки модели ***
# -----
-----

import requests
import datetime
from pyspark.sql import Row
from pandas import json_normalize

def after_all(df, spark, app, *args, **kwargs):

    LOGIN = '* * *'
    PASSWORD = '* * *'
    DOMAIN = 'https://* * *.ru'

    AUTH = {
        'Login': LOGIN,
        'Password': PASSWORD,
    }

    session = requests.Session()
    # cookies = session.cookies.get_dict() # {}

    # авторизация
    response = session.post(url=DOMAIN+'/api/auth/login', json=AUTH,
verify=False)

    cookies = session.cookies.get_dict()

    # print(cookies)

    # получение LINQ-запроса

```

```
LINQ = {
    'DataSourceKey': 'BI_milestones',
    'PageSize': 1000,
}

r = session.post(url=DOMAIN+'/api/queries/get', cookies=cookies,
json=LINQ, verify=False)

if not r.ok:
    raise Exception(f'Ошибка в {r.url}. HTTP {r.status_code}: {r.text}')

data = r.json()

#Создадим список, который послужит основой для создания DataFrame
rows = []
for line in data:
    print(line)
    rows.append(Row(
        id=line['Id'],
        name=line['Name'],
        projectid=line['ProjectId'],
        systemstartdate=line['SystemStartDate'],
        systemenddate=line['SystemEndDate'],
        actualstartdate=line['ActualStartDate'],
        actualenddate=line['ActualEndDate'],
        baselineplanstartdate=line['BaselinePlanStartDate'],
        baselineplanenddate=line['BaselinePlanEndDate'],
        bossfio=line['BossFIO'],
        rpfio=line['RPFIO'],
        description=line['Description'],
        result=line['Result'],
        statusreport_date=line['StatusReport_Date'],
        statusreport_что_sdelano_problemi_riski=line['StatusReport_Что_sdelano_problemi_riski'],
        statusreport_prichina_problemi=line['StatusReport_Prichina_problemi'],
        statusreport_statusname=line['StatusReport_StatusName']
    ))

return spark.createDataFrame(rows)
```

Символами \*\*\* в примерах скриптов обозначены автоматически формируемые данные (например, номер модели) или данные, имеющие отношение к конкретной учетной записи в ADVANTA (например, логин, пароль и домен).

Затем необходимо опубликовать скрипт и обновить модель. После обновления данные из ADVANTA автоматически загрузятся в модель.

## Пример модели

AK ID	AK NAME	AK PROJECT ID	SYSTEMSTARTDATE	SYSTEMENDDATE	ACTUALSTARTDATE	ACTUALENDDATE
dc05bb77-4114-49a3-9b15-0034eab2e...	Выполнены все мероприятия дорож...	dfa43076-c8a8-4eb5-8505-c6277535...	2024-12-25T17:00:00	2024-12-25T17:00:00		
de78aeb2-b314-4df5-a100-01088ae13...	ПНР выполнены	a6ed1995-582c-42d3-afa0-6db8c280...	2024-02-07T17:00:00	2024-02-07T17:00:00		
97700185-17a3-44a7-9394-049f5085...	Старт проекта	1787182a-e1bd-4073-8382-4eab3206...	2023-06-30T08:00:00	2023-06-30T17:00:00	2023-06-30T08:00:00	2023-06-30T17:00:00
c8aa5cdd-539f-4934-885e-061caeef...	Старт проекта	87ba6e25-b18e-471a-a079-38f38act5...	2023-08-30T08:00:00	2023-08-30T08:00:00	2023-09-03T23:11:36	2023-09-03T23:11:36
2678aeeef-bcd3-4d59-9a50-1149f79d1...	Планирование мероприятия провед...	87ba6e25-b18e-471a-a079-38f38act5...	2023-10-31T17:00:00	2023-10-31T17:00:00		
55485809-c60c-4e9c-9c10-11913a8b...	Показатели стабилизированы	c953fa6e-6901-442d-8916-ad5ec071e...	2023-10-23T17:00:00	2023-10-23T17:00:00		
caebdb58-2a16-4312-8162-1a84b73a...	Оборудование введено в эксплуатац...	e0a78994-6a06-4a84-baef-f88122c4...	2023-11-24T17:00:00	2023-11-24T17:00:00		

### 4) Создать виджеты для отображения данных и дашборды из виджетов

Для реализации этих шагов необходимо ознакомиться с возможностями системы Analytic Workspace. В этом вам помогут [база знаний](#), [обучающий курс](#) и специалисты, контакты которых вы можете найти на сайте [Analytic Workspace](#).

### Пример дашборда «Здоровье портфеля» в Analytic Workspace на основе данных, полученных из ADVANTA

Информационные панели

### Здоровье портфеля

Редактировать Клонировать Удалить

---

**Статус проекта**

Статус

ПРИЧИНА ПРОБЛЕМЫ

- Пустые
- Проблем нет
- Решу проблемы сам
- Требуется помощь

**Список проектов**

НАЗВАНИЕ	РУКОВОДИТЕЛЬ ПРОЕКТА	ЧТО СДЕЛАНО, ПРОБЛЕМЫ, РИСКИ
Система управления складом	Чернов Михаил	Необходимо 2 новых человека в команду проекта
Внедрение системы управления мотивацией сотрудников	Жиганова Людмила	Не утверждены результаты работ по устранению расхождений (план)
Черновки Организационного проекта	Воронов Олег	
Проведение командообразующего мероприятия	Жиганова Людмила	Требуется выделение дополнительных средств для дальнейшей реализации
Организация мобильной VPN	Воронов Олег	комментарий
Интеграция с JIRA	Воронов Олег	все по плану

---

**Статус КТ**

Статус

ПРИЧИНА ПРОБЛЕМЫ

- Пустые
- Проблем нет
- Решу проблемы сам
- Требуется помощь

**Список КТ**

НАЗВАНИЕ	ДАТА НАЧАЛА ПО ВП	ДАТА ЗАВЕРШЕНИЯ ПО ВП	СТАТУС	ЧТО СДЕЛАНО, ПРОБЛЕМЫ, РИСКИ
Отчет о завершении этапа 1 согласован	2023-09-05 17:00:00	2023-09-05 17:00:00	Решу проблемы сам	на рассмотрении
Мероприятие проведено	2023-10-27 17:00:00	2023-10-27 17:00:00		
Оборудование доставлено на площадку				
Согласована площадка	2023-11-02 17:00:00	2023-11-02 17:00:00		
Проект завершен	2024-04-26 17:00:00	2024-04-26 17:00:00		
Согласована площадка	2023-11-02 17:00:00	2023-11-02 17:00:00		

[Оставить заявку на демонстрацию или внедрение ADVANTA](#)

[Узнать подробнее о системе Analytic Workspace](#)

From: <https://wiki.a2nta.ru/> - Wiki [3.x]

Permanent link: [https://wiki.a2nta.ru/doku.php/product/api/integration\\_examples/aw?rev=1714742882](https://wiki.a2nta.ru/doku.php/product/api/integration_examples/aw?rev=1714742882)

Last update: **03.05.2024 13:28**

