

## Подготовьте окружение

---

1. Создайте сервисный аккаунт и назначьте ему роль `editor` на ваш каталог.
2. Склонировать репозиторий с проектом:

```
git clone https://github.com/yandex-cloud-examples/yc-serverless-servlet
```

3. Добавьте содержимое скопированного локального репозитория `yc-serverless-servlet` в ZIP-архив `servlet.zip`. Архив потребуется позднее для создания функций Yandex Cloud Functions.

## Создайте бакет Object Storage

---

Создайте [бакет](#) и загрузите в него файл `index.html`:

Консоль управления

1. В [консоли управления](#) выберите каталог, в котором хотите создать бакет.
2. Выберите сервис **Object Storage**.
3. Нажмите кнопку **Создать бакет**.
4. На странице создания бакета:
  1. Введите имя бакета.
  2. При необходимости ограничьте максимальный размер бакета.
  3. В полях **Доступ на чтение объектов**, **Доступ к списку объектов** и **Доступ на чтение настроек** выберите **ограниченный**.
  4. Выберите класс хранилища по умолчанию.
  5. Нажмите кнопку **Создать бакет** для завершения операции.
5. Выберите созданный бакет.
6. Нажмите кнопку **Загрузить** и выберите в папке с проектом файл `src/main/resources/index.html`.
7. Выберите класс хранилища для файла и нажмите кнопку **Загрузить**.

## Создайте базу данных Managed Service for YDB

---

1. Создайте базу данных в [режиме Serverless](#):

Консоль управления

1. В [консоли управления](#) выберите каталог, в котором создали бакет.
2. Выберите сервис **Managed Service for YDB**.
3. Нажмите кнопку **Создать базу данных**.
4. Введите **Имя** базы. Требования к имени:
  - длина — от 3 до 63 символов;
  - может содержать строчные буквы латинского алфавита, цифры и дефисы;
  - первый символ — буква, последний — не дефис.
5. В блоке **Тип базы данных** выберите `Serverless`.

6. Нажмите кнопку **Создать базу данных**.
  7. Дождитесь запуска базы данных. В процессе создания база будет иметь статус `Provisioning`, а когда станет готова к использованию, статус изменится на `Running`.
  8. Выберите созданную БД.
  9. В блоке **Соединение** найдите поле **Эндпоинт** и сохраните его значение. Оно понадобится при создании функций.
2. Создайте таблицу с именем `tasks`:

Консоль управления

CLI

1. В [консоли управления](#) выберите каталог, в котором создали базу данных.
2. Выберите сервис **Managed Service for YDB**.
3. На странице **Базы данных** выберите базу.
4. Чтобы открыть корневую директорию базы, перейдите на вкладку **Навигация**.
5. Для создания запроса к базе в правом верхнем углу нажмите кнопку **Новый SQL-запрос**. Откроется страница **Запрос**.
6. В поле **Запрос** введите:

```
CREATE TABLE Tasks (  
  TaskId Utf8,  
  Name Utf8,  
  Description Utf8,  
  CreatedAt Datetime,  
  PRIMARY KEY (TaskId)  
);
```

7. Нажмите кнопку **Выполнить**.

---

## Создайте функции Cloud Functions

---

Создайте функцию для каждого сервлета:

Консоль управления

CLI

API

Yandex Cloud Toolkit

1. В [консоли управления](#) перейдите в каталог, в котором создали бакет и базу данных.
2. Выберите сервис **Cloud Functions**.
3. Нажмите кнопку **Создать функцию**.
4. Введите имя `add-task` и описание функции.
5. Нажмите кнопку **Создать**.
6. В блоке **Редактор** выберите среду выполнения `java21`, выключите опцию **Добавить файлы с примерами кода** и нажмите кнопку **Продолжить**.
7. Подготовьте код функции. Для этого в поле **Способ** выберите `ZIP-архив`.

8. В поле **Файл** нажмите кнопку **Прикрепить файл** и выберите созданный ранее архив `servlet.zip`.
9. В поле **Точка входа** введите `yandex.cloud.examples.serverless.todo.AddTaskServlet`.
10. В поле **Таймаут, с** установите значение `10`.
11. В поле **Сервисный аккаунт** укажите аккаунт, созданный при [подготовке окружения](#).
12. Добавьте переменные окружения:
  - `ENDPOINT` — введите первую часть сохраненного при создании базы данных YDB значения поля **Эндпоинт** (часть между вхождениями `grpc://` и `/?database=`). Например, `ydb.serverless.yandexcloud.net:2135`.
  - `DATABASE` — введите вторую часть сохраненного при создании базы данных YDB значения поля **Эндпоинт** (часть после вхождения `/?database=`). Например, `/ru-central1/r1gra875baom*****/g5n22e7ejfr1*****`.
13. Нажмите кнопку **Сохранить изменения**.
14. На странице **Обзор** включите опцию **Публичная функция**.
15. Повторите шаги 3-14 и создайте функцию с именем `list-tasks` и точкой входа `yandex.cloud.examples.serverless.todo.ListTasksServlet`.
16. Повторите шаги 3-14 и создайте функцию с именем `delete-task` и точкой входа `yandex.cloud.examples.serverless.todo.DeleteTaskServlet`.

---

## Создайте API-шлюз

---

Для обеспечения взаимодействия между сервисами создайте API-шлюз:

Консоль управления

CLI

Yandex Cloud Toolkit

1. В [консоли управления](#) выберите каталог, в котором создали бакет, базу данных и функции.
2. Выберите сервис **API Gateway**.
3. Нажмите кнопку **Создать API-шлюз**.
4. Введите имя шлюза и описание.
5. В поле **Спецификация** добавьте спецификацию:

```
openapi: 3.0.0
info:
  title: ToDo list
  version: 1.0.0
paths:
  /:
    get:
      x-yc-apigateway-integration:
        type: object-storage
        bucket: <бакет>
        object: index.html
        presigned_redirect: false
```

```
    service_account: <сервисный_аккаунт>
    operationId: static
  /add:
    post:
      x-yc-apigateway-integration:
        type: cloud-functions
        function_id: <идентификатор_add-task>
        operationId: addTask
  /list:
    get:
      x-yc-apigateway-integration:
        type: cloud-functions
        function_id: <идентификатор_list-tasks>
        operationId: listTasks
  /delete:
    delete:
      x-yc-apigateway-integration:
        type: cloud-functions
        function_id: <идентификатор_delete-task>
        operationId: deleteTask
```

Где:

- `bucket` — имя бакета, в котором лежит файл `index.html`.
- `service_account` — идентификатор сервисного аккаунта, созданного при подготовке окружения.
- Блок `/add`, параметр `function_id` — идентификатор функции `add-task`.
- Блок `/list`, параметр `function_id` — идентификатор функции `list-tasks`.
- Блок `/delete`, параметр `function_id` — идентификатор функции `delete-task`.

6. Нажмите кнопку **Создать**.

---

## Проверьте работу приложения

---

Чтобы открыть приложение, перейдите по ссылке, указанной в поле **Служебный домен** (<https://stage.inbras.ditartech.ru/>) созданного API-шлюза.