

Bank-Demo-API on OPENSIFT - Rest API для демонстрации работы гибридного облака

- 1. [Цель разработки](#)
- 2. [Составные части API](#)
- 3. [Параметризация приложения](#)
- 4. [Запуск приложения локально](#)
- 5. [Запуск приложения на OpenShift](#)

Цель разработки

Этот проект демонстрирует размещение API в среде OPENSIFT. API работает с базами данных postgres и couch-db в среде OPENSIFT

Составные части API

API по работе с клиентами банка

- \server\services\service-pgdb-cust.js имплементация логики API
- \server\routers\customer_router.js имплементация http роутера
- \test\test-customer-router.js тест-кейсы при локальном запуске приложения
- \test\test-customer-router-OKD.js тест-кейсы приложения на OpenShift
- \test\test-customer-router-apigw.js тест-кейсы через IBM Cloud API-GateWay

API по работе со счетами банка

- \server\services\service-pgdb-acnt.js имплементация логики API
- \server\routers\account_router.js имплементация http роутера
- \test\test-accounts-router.js тест-кейсы при локальном запуске приложения
- \test\test-accounts-router-OKD.js тест-кейсы приложения на OpenShift
- \test\test-accounts-router-apigw.js тест-кейсы через IBM Cloud API-GateWay

API по работе с карточками банка

- \server\services\service-pgdb-cards.js имплементация логики API
- \server\routers\card_router.js имплементация http роутера
- \test\test-cards-router.js тест-кейсы при локальном запуске приложения
- \test\test-cards-router-OKD.js тест-кейсы приложения на OpenShift
- \test\test-cards-router-apigw.js тест-кейсы через IBM Cloud API-GateWay

API по регистрации визита в отделение

- \server\services\service-couchdb-appon.js имплементация логики API
- \server\routers\appon_router.js имплементация http роутера
- \test\test-appon-router.js тест-кейсы при локальном запуске приложения
- \test\test-appon-router-OKD.js тест-кейсы приложения на OpenShift
- \test\test-appon-router-apigw.js тест-кейсы через IBM Cloud API-GateWay

API по регистрации заявки на выдачу кредита

- \server\services\service-couchdb-loan.js имплементация логики API
- \server\routers\loan_router.js имплементация http роутера
- \test\test-loan-router.js тест-кейсы при локальном запуске приложения
- \test\test-loan-router-OKD.js тест-кейсы приложения на OpenShift
- \test\test-loan-router-apigw.js тест-кейсы через IBM Cloud API-GateWay

swagger - файлы для Rest API расположены в папке: ../API

Параметризация приложения

- Настройка порта Настройка порта храниться в файле \server\config\local.json По умолчанию слушается порт 8080

```
{
  "port": 8080
}
```

- Настройка переменных среды Настройка переменных среды описана в файле \server\config\mapping.json

```
{
  "version": 1,
  // Host  БД Postgres
  "DB_HOST": {
    "searchPatterns": [
      "env:DB_HOST", // <---- env  переменная при запуске в контейнере
openshift
      "file:/server/localdev-config.json:$.DB_HOST" // <---- параметр из файла
localdev-config.json при запуске локально
    ]
  },
  // Порт БД Postgres
  "DB_PORT": {
    "searchPatterns": [
      "env:DB_PORT",
      "file:/server/localdev-config.json:$.DB_PORT"
    ]
  },
  //Имя  БД Postgres
  "DB_NAME": {
    "searchPatterns": [
      "env:DB_NAME",
      "file:/server/localdev-config.json:$.DB_NAME"
    ]
  },
  // Пользователь БД Postgres для подключения к БД
  "DB_USER": {
```

```
    "searchPatterns": [
      "env:DB_USER",
      "file:/server/localdev-config.json:$.DB_USER"
    ]
  },
  // Пароль пользователя БД Postgres для подключения к БД
  "DB_PSW": {
    "searchPatterns": [
      "env:DB_PSW",
      "file:/server/localdev-config.json:$.DB_PSW"
    ]
  },
  // Host БД couchdb
  "DBC_HOST": {
    "searchPatterns": [
      "env:DBC_HOST",
      "file:/server/localdev-config.json:$.DBC_HOST"
    ]
  },
  // Порт БД couchdb
  "DBC_PORT": {
    "searchPatterns": [
      "env:DBC_PORT",
      "file:/server/localdev-config.json:$.DBC_PORT"
    ]
  },
  // Протокол для обращения к БД couchdb (http, https)
  "DBC_PROT": {
    "searchPatterns": [
      "env:DBC_PROT",
      "file:/server/localdev-config.json:$.DBC_PROT"
    ]
  },
  // Имя БД couchdb
  "DBC_NAME": {
    "searchPatterns": [
      "env:DBC_NAME",
      "file:/server/localdev-config.json:$.DBC_NAME"
    ]
  },
  // Логин пользователя БД couchdb
  "DBC_USERNAME": {
    "searchPatterns": [
      "env:DBC_USERNAME",
      "file:/server/localdev-config.json:$.DBC_USERNAME"
    ]
  },
  // Пароль пользователя БД couchdb
  "DBC_PSW": {
    "searchPatterns": [
      "env:DBC_PSW",
      "file:/server/localdev-config.json:$.DBC_PSW"
    ]
  }
}
```

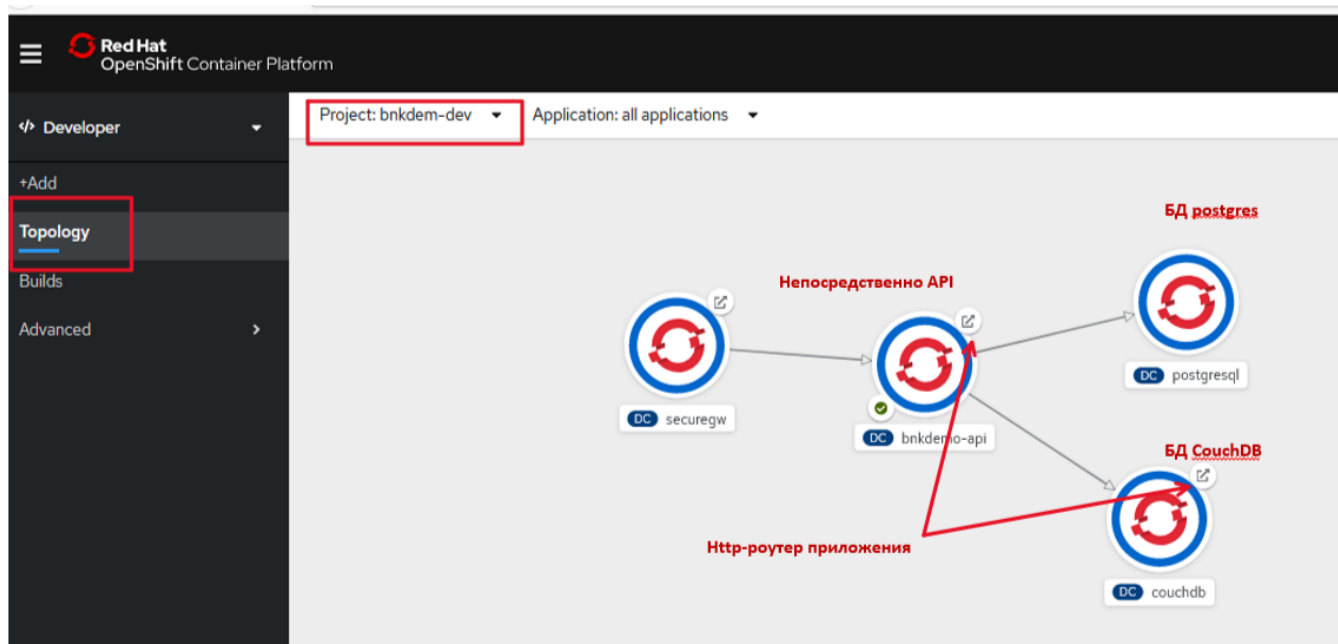
```
}
```

При запуске локально необходимо настроить параметры подключения к базам данных из файла `server/localdev-config.json`. Файл не попадает в source control - чтобы не передавать локальные секретные настройки. Пример файла показан ниже:

```
{
  "DB_HOST": "localhost",
  "DB_USER": "*****",
  "DB_PSW": "*****",
  "DB_NAME": "*****",
  "DB_PORT": "*****",
  "DBC_HOST": "*****",
  "DBC_PROT": "http",
  "DBC_PORT": 80,
  "DBC_NAME": "*****",
  "DBC_USERNAME": "*****",
  "DBC_PSW": "*****"
}
```

Запуск приложения локально

- Выполнить настройки на базы данных couch db и postgres db. Предполагается, что базы данных уже задеплоены на openshift.



pic-1

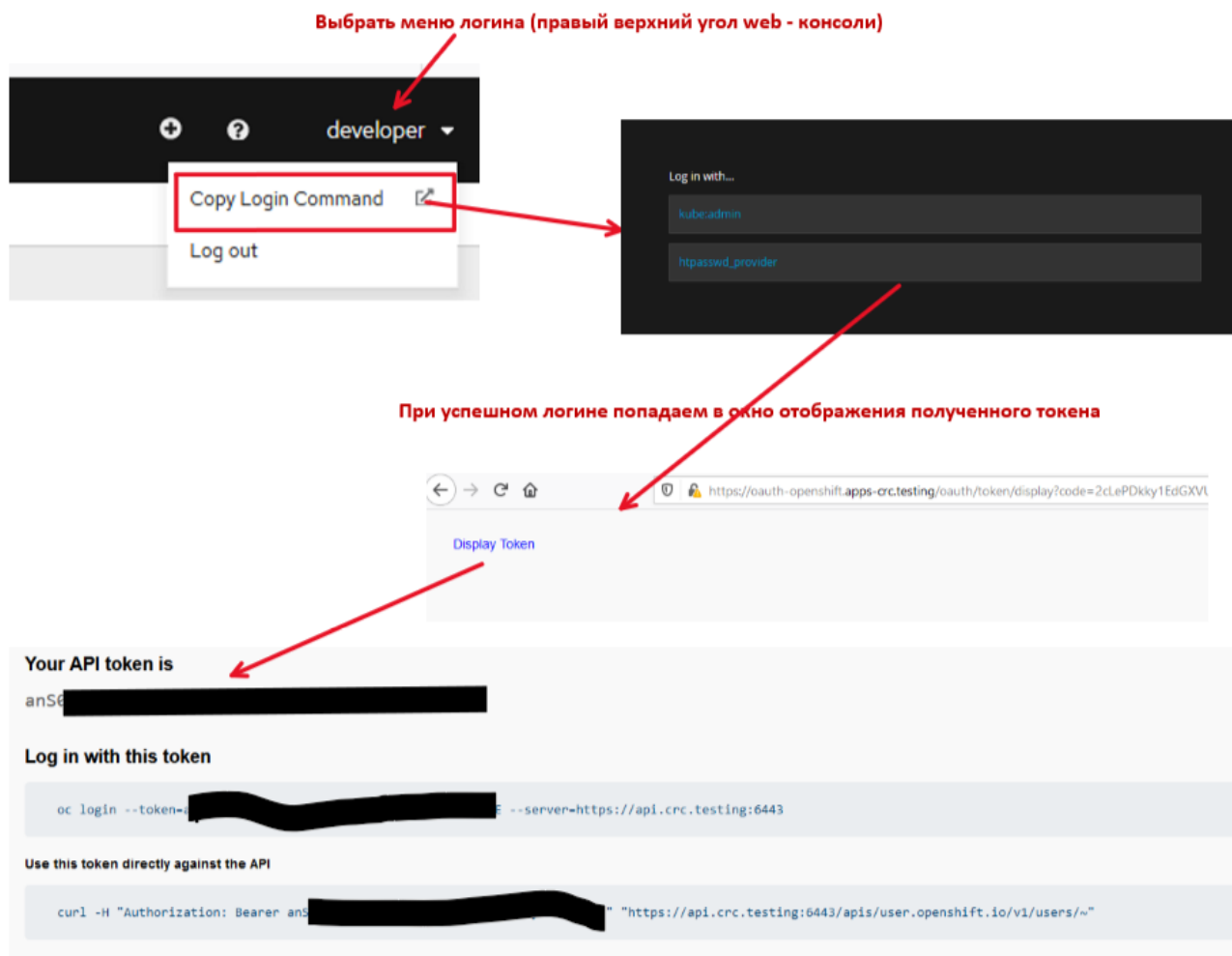
БД postgres работает по собственному протоколу. Поэтому, нужно "пробросить" порт pod-a, на котором запущена БД на локальную станцию. Для этого:

- запустить powershell (bash)

- зайти в openshift для этого выполнить команду OpenShift CLI

```
oc login --token=your token --server=openshift api url
```

команду можно скопировать, сгенерировав token доступа



pic-2

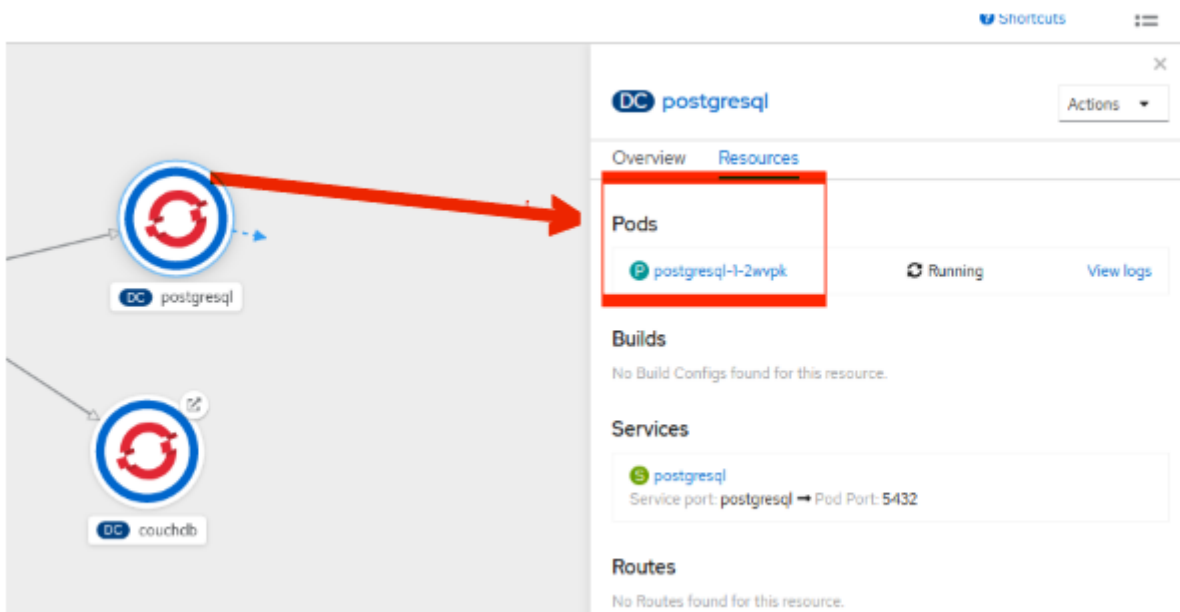
- выбрать проект

```
oc project YourProjectName
```

- выполнить команду проброса порта

```
oc port-forward postgresql-1-2wvpk 15432:5432
```

postgresql-1-2wvpk - имя пода



pic-3

15432 - номер локального порта рабочей станции 5432 - номер порта pod в OpenShift

- Убедиться, что локальный порт указан в настройках файла server/localdev-config.json, переменная DB_PORT.

Результат работы команд показан ниже

```
PS C:\Users\PavloShcherbukha> oc login --token=**** --server=https://*****
Logged into "https://api.crc.testing:6443" as "developer" using the token
provided.
```

You have access to the following projects and can switch between them with 'oc project <projectname>':

```
bnk-dev
* bnkdem-dev
bnkdem-int
bnkdem-prod
```

Using project "bnkdem-dev".

```
PS C:\Users\PavloShcherbukha> oc port-forward postgresql-1-2wvpk 15432:5432
Forwarding from 127.0.0.1:15432 -> 5432
Forwarding from [::1]:15432 -> 5432
Handling connection for 15432
Handling connection for 15432
Handling connection for 15432
Handling connection for 15432
Handling connection for 15432
Handling connection for 15432
Handling connection for 15432
```

```
Handling connection for 15432
Handling connection for 15432
```

Запустить команды:

```
npm install
npm start
```

При успешном запуске должны успешно отработать тестовые кейсы при локальном запуске приложения [Составные части API](#)

Запуск приложения на OpenShift

Для запуска приложения на Openshift необходимо выполнить шаги deployments