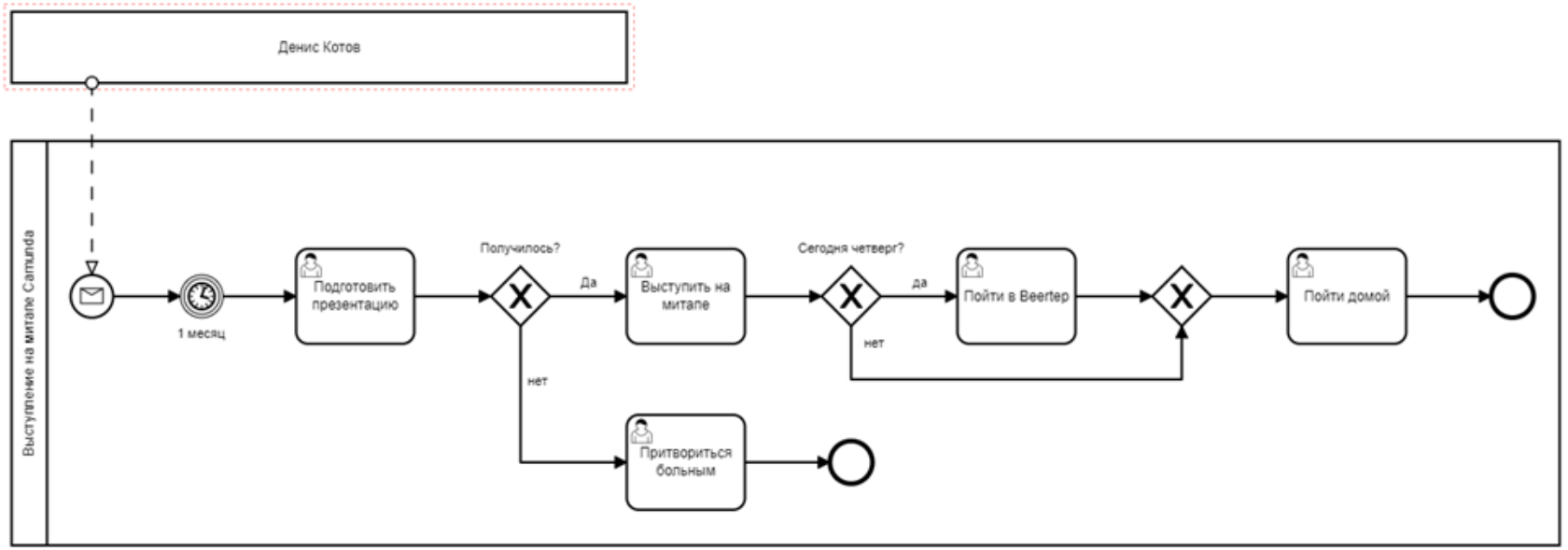
Как избавиться от контекста и начать жить

Tinkoff.ru

План

- Контекст процесса в Camunda
- Проблемы
- Выводы
- Пример

Контекст процесса в Camunda



Контекст процесса в Camunda

Instance variables (max 150)

- presentation
- ready
- time
- spentTime
- participants

```
{
  type: "Object",
  value: {
    slides: [
      "slide#1",
      "slide#2",
      "slide#3"
    ],
    name: "Как избавиться от контекста и начать жить",
    author: {
      name: "Евгений Изместьев"
```

Контекст процесса в Camunda

presentation
ready
time
spentTime
participants


```
{
  type: "Object",
  value: {
    slides: [
      "slide#1",
      "slide#2",
      "slide#3"
    ],
    name: "Как избавиться от контекста и начать жить",
    author: {
      name: "Евгений Изместьев"
    }
  },
  valueInfo: {
    objectTypeName: "ru.tinkoff.bpm.smecreditorigination.process.presentation.Presentation",
    serializationDataFormat: "application/x-java-serialized-object"
  },
  id: "84c0f57f-5952-11ea-9705-e0d55e43e2d1",
  name: "presentation",
  processDefinitionKey: "CamundaPresentation",
  processDefinitionId: "CamundaPresentation:1:1bafcb0d-5951-11ea-a761-e0d55e43e2d1",
  processInstanceId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1",
  executionId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1",
  activityInstanceId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1",
  caseDefinitionKey: null,
  caseDefinitionId: null,
  caseInstanceId: null,
  caseExecutionId: null,
  taskId: null,
  errorMessage: null,
  tenantId: null,
  state: "CREATED",
  createTime: "2020-02-27T14:15:53.771+0300",
  removalTime: null,
  rootProcessInstanceId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1"
}
```

Контекст процесса в Camunda

presentation

ready

time

spentTime

participants


```
{
  type: "Boolean",
  value: true,
  valueInfo: {
  },
  id: "84c11c90-5952-11ea-9705-e0d55e43e2d1",
  name: "ready",
  processDefinitionKey: "CamundaPresentation",
  processDefinitionId: "CamundaPresentation:1:1bafcb0d-5951-11ea-a761-e0d55e43e2d1",
  processInstanceId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1",
  executionId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1",
  activityInstanceId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1",
  caseDefinitionKey: null,
  caseDefinitionId: null,
  caseInstanceId: null,
  caseExecutionId: null,
  taskId: null,
  errorMessage: null,
  tenantId: null,
  state: "CREATED",
  createTime: "2020-02-27T14:15:53.771+0300",
  removalTime: null,
  rootProcessInstanceId: "84bf6edd-5952-11ea-9705-e0d55e43e2d1"
}
```

Контекст процесса в Camunda

Data Output | Explain | Messages | Notifications

	id_ [PK] character varying (64)	rev_ integer	type_ character varying (255)	name_ character varying (255)	execution_id_ character varying (64)	proc_inst_id_ character varying (64)	case_execu character v
1	9c787c64-5957-11ea-9705-e0d55...	1	serializable	presentation	9c785552-5957-11ea-9705-...	9c785552-5957-11ea-9705-...	[null]
2	9c787c65-5957-11ea-9705-e0d55...	1	boolean	ready	9c785552-5957-11ea-9705-...	9c785552-5957-11ea-9705-...	[null]
3	9c787c67-5957-11ea-9705-e0d55...	1	serializable	time	9c785552-5957-11ea-9705-...	9c785552-5957-11ea-9705-...	[null]
4	9c787c68-5957-11ea-9705-e0d55...	1	integer	spentTime	9c785552-5957-11ea-9705-...	9c785552-5957-11ea-9705-...	[null]
5	9c787c6a-5957-11ea-9705-e0d55...	1	serializable	participants	9c785552-5957-11ea-9705-...	9c785552-5957-11ea-9705-...	[null]

Проблемы

Проблема №1: Одна таблица

```
SELECT RES.*, (
    case
        when RES.TASK_ID_ is not null and RES.EXECUTION_ID_ is not null
        then EXECUTION.ACT_INST_ID_
        when RES.CASE_EXECUTION_ID_ is not null
        then RES.CASE_EXECUTION_ID_
        when EXECUTION.PARENT_ID_ is null and RES.IS_CONCURRENT_LOCAL_ = 0
        then EXECUTION.ID_
        when EXECUTION.IS_SCOPE_ = 1 and EXECUTION.PARENT_ID_ is not null and RES.IS_CONCURRENT_LOCAL_ = 0
        then PARENT_EXECUTION.ACT_INST_ID_
        else EXECUTION.ACT_INST_ID_
    end
) ACT_INST_ID_
    FROM ACT_RU_VARIABLE RES
    LEFT JOIN ACT_RU_EXECUTION EXECUTION ON RES.EXECUTION_ID_ = EXECUTION.ID_
    LEFT JOIN ACT_RU_EXECUTION PARENT_EXECUTION ON EXECUTION.PARENT_ID_ = PARENT_EXECUTION.ID_
    WHERE EXECUTION_ID_ = 9c785552-5957-11ea-9705-e0d55e43e2d1 AND TASK_ID_ is null
```

Проблемы

Проблема №2: Сериализация

```
data class Presentation (
    val id: Long,
    val name: String,
    val slides: List<Slide>
) : Serializable
```

Проблемы

Проблема №2: Сериализация

```
data class Presentation (
    val id: Long,
    val name: String,
    val slides: List<Slide>
) : Serializable
```

Проблемы

Проблема №2: Сериализация

```
data class Presentation (
    val id: Long,
    val name: String,
    val slides: List<Slide>,
    val date: LocalDate
) : Serializable
```

Проблемы

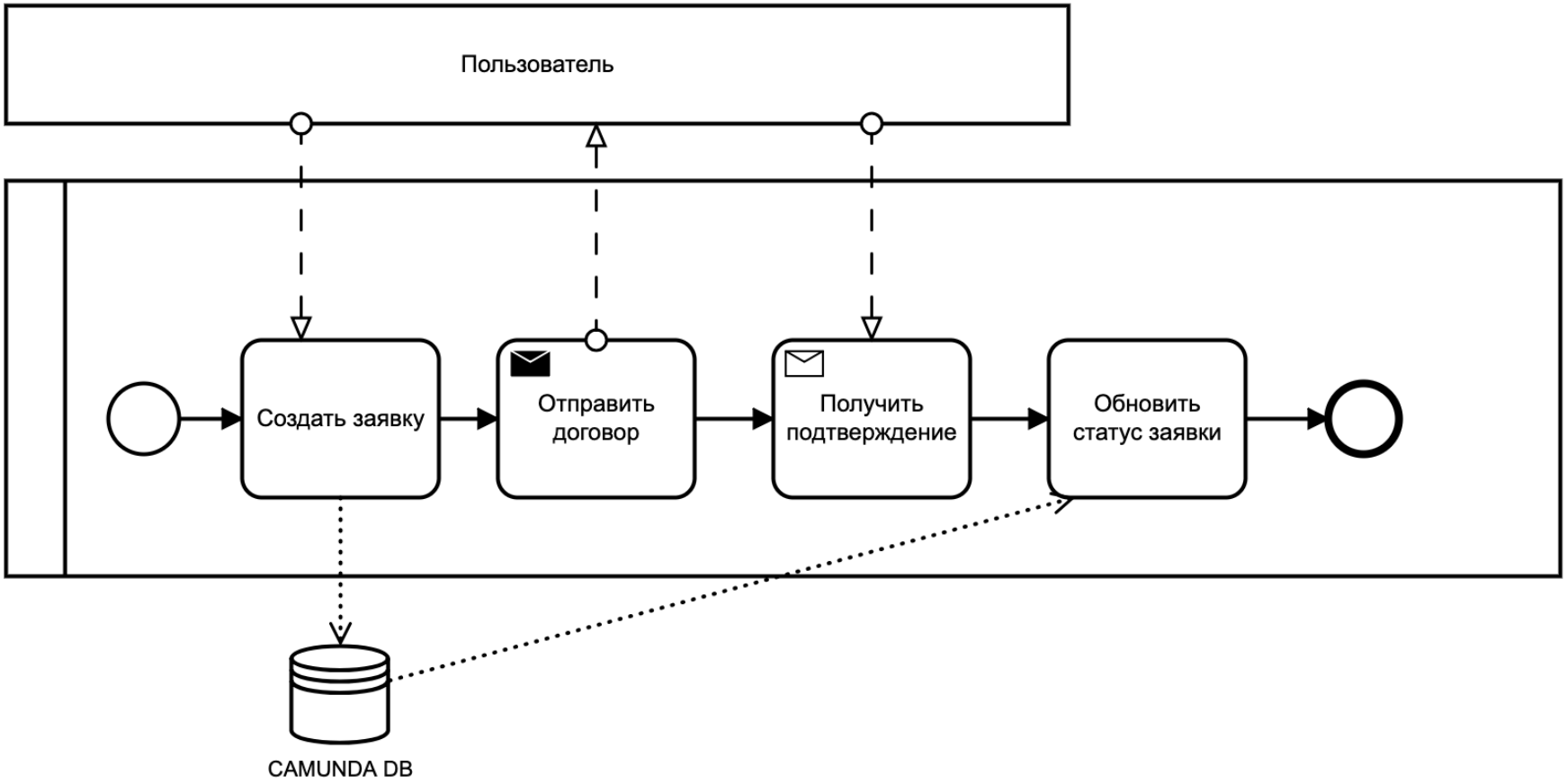
Проблема №2: Сериализация

```
data class Presentation (
    val id: Long,
    val name: String,
    val slides: List<Slide>,
    val date: LocalDate
) : Serializable  {

    companion object {
        private val serialVersionUID = 1L

    }
}
```

Проблемы

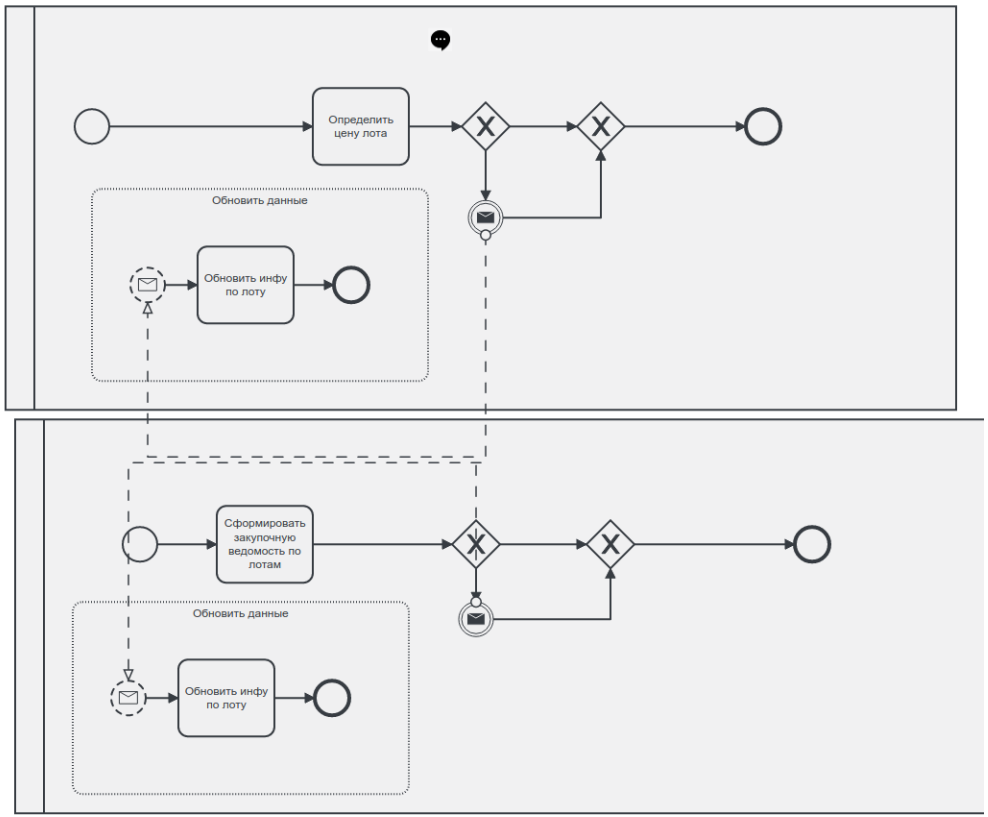
Проблема №3:

Связь модели данных и сериализованных данных в DB



Проблемы

Проблема №4: N-процессов



Проблемы

Проблема №5: Сложный CRUD

Example

Request

POST `/execution/anExecutionId/localVariables`

Request Body:

```
{"modifications":
    {"aVariable": {"value": "aValue", "type": "String"},
    "anotherVariable": {"value": 42, "type": "Integer"}},
"deletions": [
    "aThirdVariable", "FourthVariable"
]}
```

Response

Status 204. No content.

Проблемы

Проблема №5: Сложный CRUD

```
{
  "type": "Object",
  "value": {
    "slides": [
      "slide#1",
      "slide#2",
      "slide#3"
    ],
    "name": "Как избавиться от контекста и начать жить",
    "author": {
      "name": "Евгений Изместьев"
    }
  }
}
```

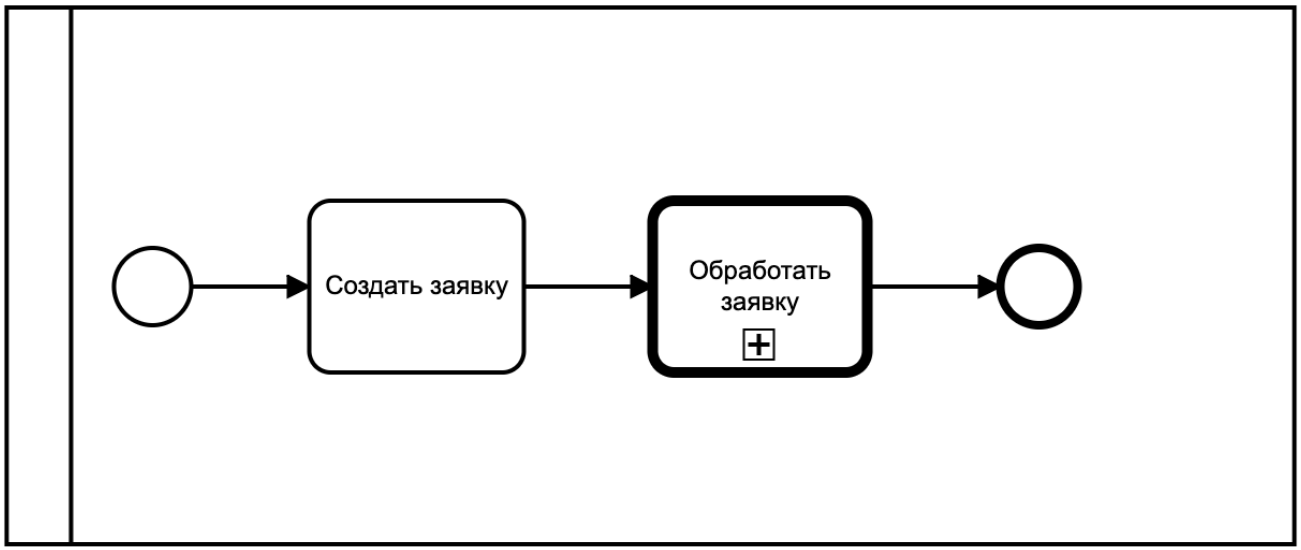
Проблемы

Проблема №5: Сложный CRUD

```
{
  "modifications": {
    "presentation": {
      "type": "Object"
      "value": "{ \"slides\": [\"slide#1\",\"slide#2\",\"slide#3\"],\"name\": \"Как избавиться от контекста и
        начать жить\",\"author\": {\"name\": \"Денис Котов\"}}"
      "valueInfo": {
        "objectTypeName": "ru.tinkoff.bpm.camunda.sme.model.dto.Presentation",
        "serializationDataFormat": "application/json"
      }
    },

  }
}
```

Проблемы

Проблема №6: Передача переменных в call activity



Проблемы

- Проблема №6: Передача переменных в call activity
- Решение: Попробовать писать тесты на схему

Проблемы

- Проблема №5: Сложный CRUD
- Решение: Попробовать декомпозировать модели

Проблемы

- Проблема №4: N-процессов
- Решение: Попробовать не делать таких процессов

Проблемы

- Проблема №3:

  Связь модели данных и сериализованных данных в БД

- Решение: Попробовать не забывать про эту проблему

Проблемы

- Проблема №2: Сериализация
- Решение: Попробовать не забывать про сериализацию

Проблемы

- Проблема №1: Одна таблица

- Решение: Попробовать вынести контекст процесса в отдельное хранилище

Варианты

- Вариант №1: Использовать Camunda DB
- Вариант №2: Использовать свою DB
- Вариант №3: Вынести в отдельное приложение

Плюсы

- Забыть о проблемах сериализации
- Свой CRUD для управление контекстом
- Контроль над DB
- Любая модель хранения

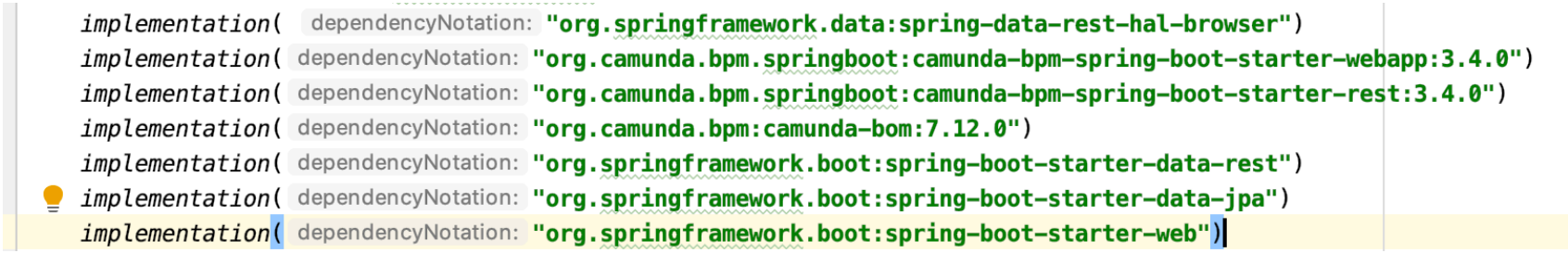
Минусы

- Отдельная DB (ресурсы)
- Контроль транзакции
- Трудоёмко на старте
- Сложнее тестировать
- Выше порог входа

Пример

- Вариант №1: Использовать Camunda DB

Пример


implementation(dependencyNotation: "org.springframework.data:spring-data-rest-hal-browser")
implementation(dependencyNotation: "org.camunda.bpm.springboot:camunda-bpm-spring-boot-starter-webapp:3.4.0")
implementation(dependencyNotation: "org.camunda.bpm.springboot:camunda-bpm-spring-boot-starter-rest:3.4.0")
implementation(dependencyNotation: "org.camunda.bpm:camunda-bom:7.12.0")
implementation(dependencyNotation: "org.springframework.boot:spring-boot-starter-data-rest")
implementation(dependencyNotation: "org.springframework.boot:spring-boot-starter-data-jpa")
implementation(dependencyNotation: "org.springframework.boot:spring-boot-starter-web")

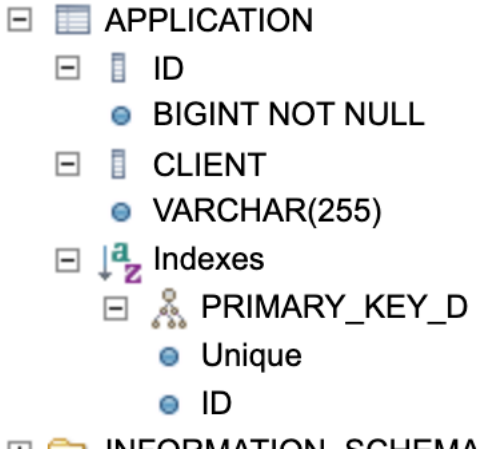
Пример

```
package com.example.demo

import org.springframework.boot.autoconfigure.SpringBootApplication
import org.springframework.boot.runApplication


@SpringBootApplication
class DemoApplication

fun main(args: Array<String>) {
  runApplication<DemoApplication>(*args)
}
```

Пример

```
@Entity
@Table(name = "application")
data class Application(
    @Id
    @GeneratedValue
    var id: Long? = null,
    var client: String? = null
)
```

Пример



Run | Run Selected | Auto complete | Clear | SQL statement:

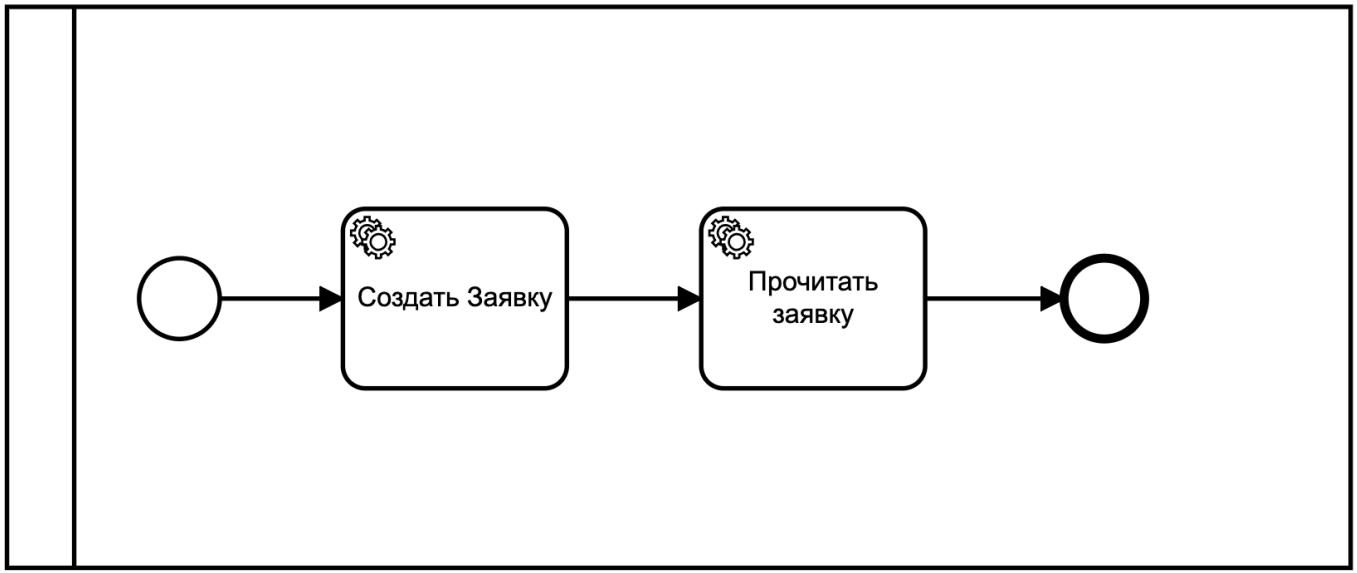
```
SELECT * FROM APPLICATION
```

@edit SELECT * FROM APPLICATION;

Action	ID	CLIENT
	1	*null*
	2	rest client 1

(2 rows, 1 ms)

Пример



Пример

```
@Component
class ApplicationCreatorDelegate(
    private val repo: ApplicationRepo
) : JavaDelegate {
    override fun execute(execution: DelegateExecution) {
        val application = repo.save(Application())
        execution.setVariable("appId", application.id)
    }
}
```

Пример

```
@Component
class ApplicationCreatorDelegate(
    private val repo: ApplicationRepo
) : JavaDelegate {
    override fun execute(execution: DelegateExecution) {
        val application = repo.save(Application())
        execution.setVariable("appId", application.id)
    }
}
```

Пример

```
@Component
class ApplicationCreatorDelegate(
    private val repo: ApplicationRepo
) : JavaDelegate {
    override fun execute(execution: DelegateExecution) {
        val application = repo.save(Application())
        execution.setVariable("appId", application.id)
    }
}
```

Пример

```
@Component
class ApplicationReaderDelegate(
    private val repo: ApplicationRepo
) : JavaDelegate {

    override fun execute(execution: DelegateExecution) {
        val appId = execution.getVariable("appId") as Long
        val application = repo.findById(appId)
        logger.info { "application = $application" }
    }
}
```

Пример

```
@Component
class ApplicationReaderDelegate(
    private val repo: ApplicationRepo
) : JavaDelegate {

    override fun execute(execution: DelegateExecution) {
        val appId = execution.getVariable("appId") as Long
        val application = repo.findById(appId)
        logger.info { "application = $application" }
    }
}
```

Пример

```
@Component
class ApplicationReaderDelegate(
    private val repo: ApplicationRepo
) : JavaDelegate {

    override fun execute(execution: DelegateExecution) {
        val appId = execution.getVariable("appId") as Long
        val application = repo.findById(appId)
        logger.info { "application = $application" }
    }
}
```

Пример

```
@RepositoryRestResource(collectionResourceRel = "application", path = "application")
interface ApplicationRepo : PagingAndSortingRepository<Application, Long> {

    override fun findById(@Param("id") name: Long): Optional<Application>
}
```

Пример

```
@RepositoryRestResource(collectionResourceRel = "application", path = "application")
interface ApplicationRepo : PagingAndSortingRepository<Application, Long> {

    override fun findById(@Param("id") name: Long): Optional<Application>
}
```

Пример

The HAL Browser (for Spring Data REST) Go To Entry Point About The HAL Browser (for Spring Data REST)

Explorer

http://localhost:8080/application/2 Go!

Custom Request Headers

Properties

```
{
  "client": "rest client 1"
}
```

Links

rel	title	name / index	docs	GET	NON-GET
self				→	!
application				→	!

Inspector

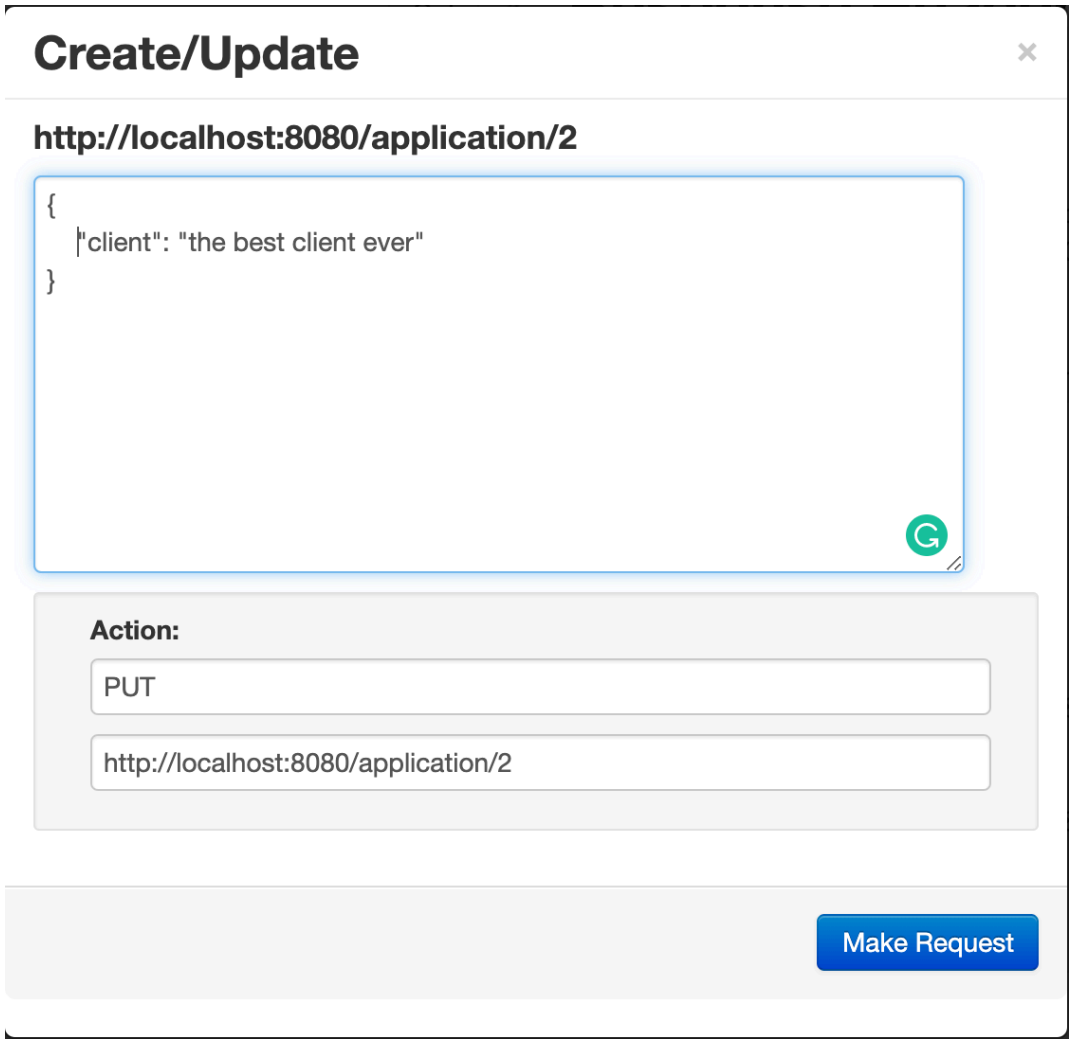
Response Headers

```
200 success

connection: keep-alive
content-type: application/hal+json
date: Thu, 27 Feb 2020 06:54:10 GMT
keep-alive: timeout=60
location: http://localhost:8080/application/2
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
```

Response Body

```
{
  "client": "rest client 1",
  "_links": {
    "self": {
      "href": "http://localhost:8080/application/2"
    },
    "application": {
      "href": "http://localhost:8080/application/2"
    }
  }
}
```

Пример


Create/Update
http://localhost:8080/application/2
{
"client": "the best client ever"
}
Action:
PUT
http://localhost:8080/application/2
Make Request

Пример

Custom Request Headers

Properties

```
{
  "client": "the best client ever"
}
```

Links

rel	title	name / index	docs	GET	NON-GET
self				➡	!
application				➡	!

Тинькофф

Дальше действовать будем мы!

Tinkoff.ru