Федеральное государственное бюджетное образовательное учреждение   
высшего образования

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | «Московский государственный технический университет  им. Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) |

ФАКУЛЬТЕТ – Информатика и управления

КАФЕДРА – Информационные системы и телекоммуникации

**РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе на тему

Разработка программного обеспечения.

Разработка OSGi микро-сервиса графического языка с моделью типовых представлений (типовых диаграмм)

Студент группы ИУ3-72 (подпись) 19.12.2016 М.А. Сидоренко

Руководитель курсовой работы (подпись) 22.12.2016 А.М. Иванов

Москва, 2016

Целью курсовой работы была разработка OSGi микро-сервиса графического языка с моделью типовых представлений (типовых диаграмм). Проект разрабатывался в среде Eclipse, а его тест был проведен в Google-приложении Postman. На рисунках 1-6 показаны результаты работы.

В приложении 1 представлен код, где вызывается метод SideBar, а соответственно в приложении 2 представлен сам класс SideBar.

В приложении 3 представлен класс Diagram, где описывается три различных типа диаграмм.

В приложении 4 код, интерфейса приложения.

Вызов диаграмм осуществляется с помощью JSON.

Рисунок 1. Вид проекта в среде разработки Eclipse.

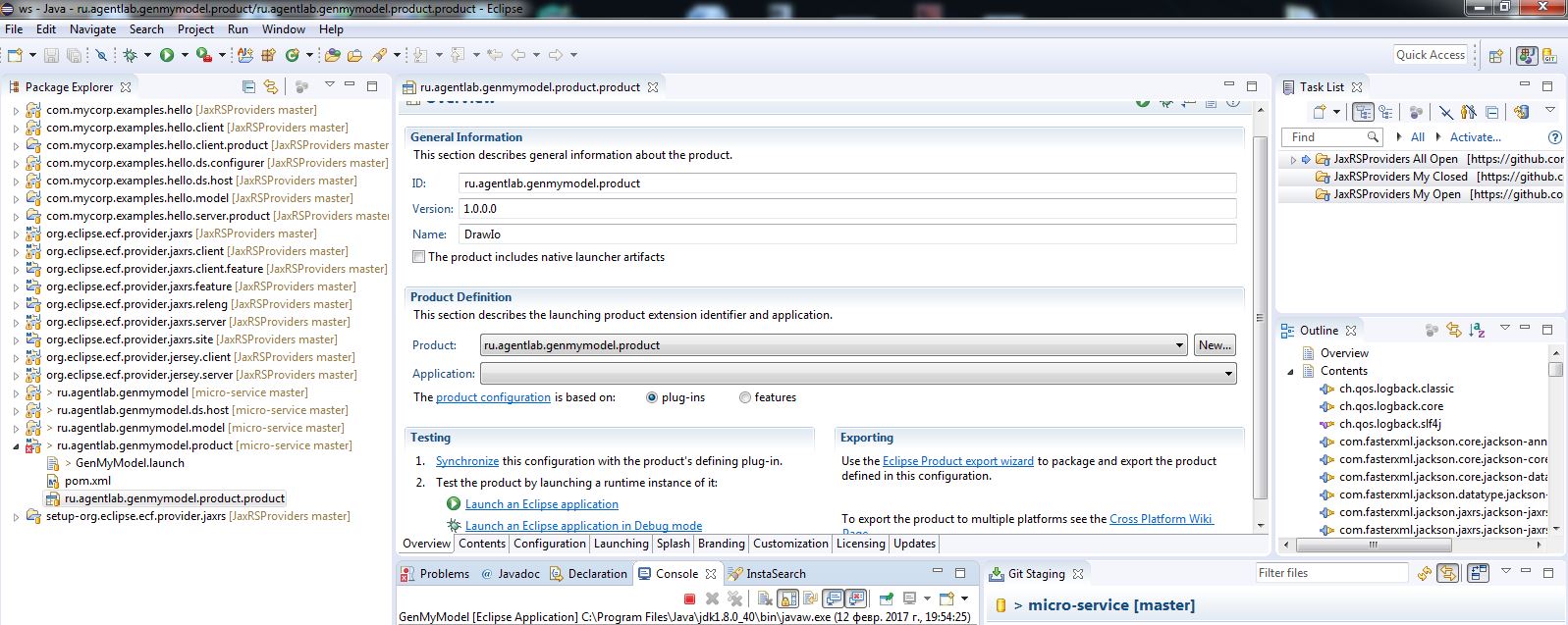


Рис 2. Вид проектав Google-приложении Postman.

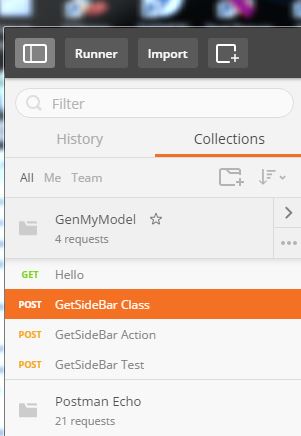


Рис 3. Вызов классовой диаграмы.

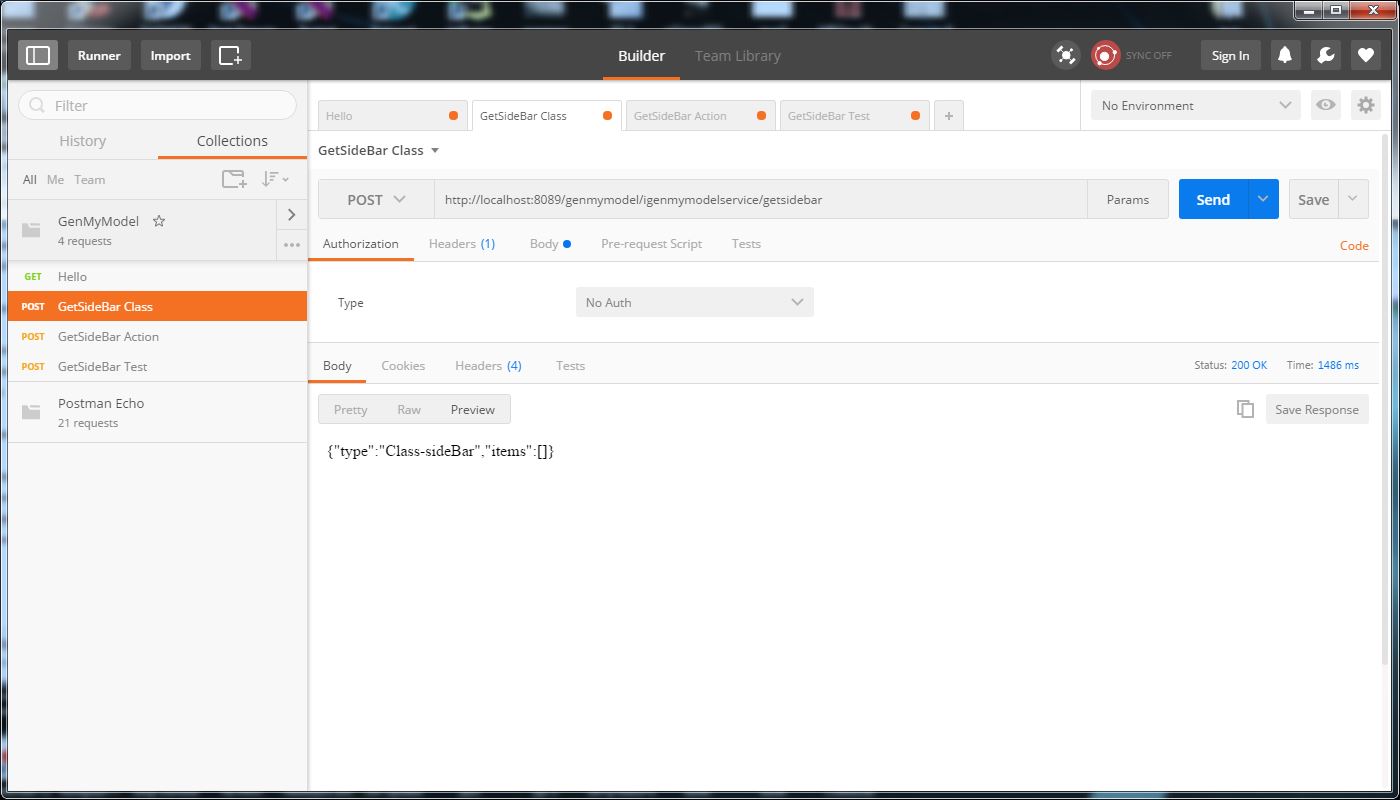


Рис 4. Активная диаграмма до ее вызова.

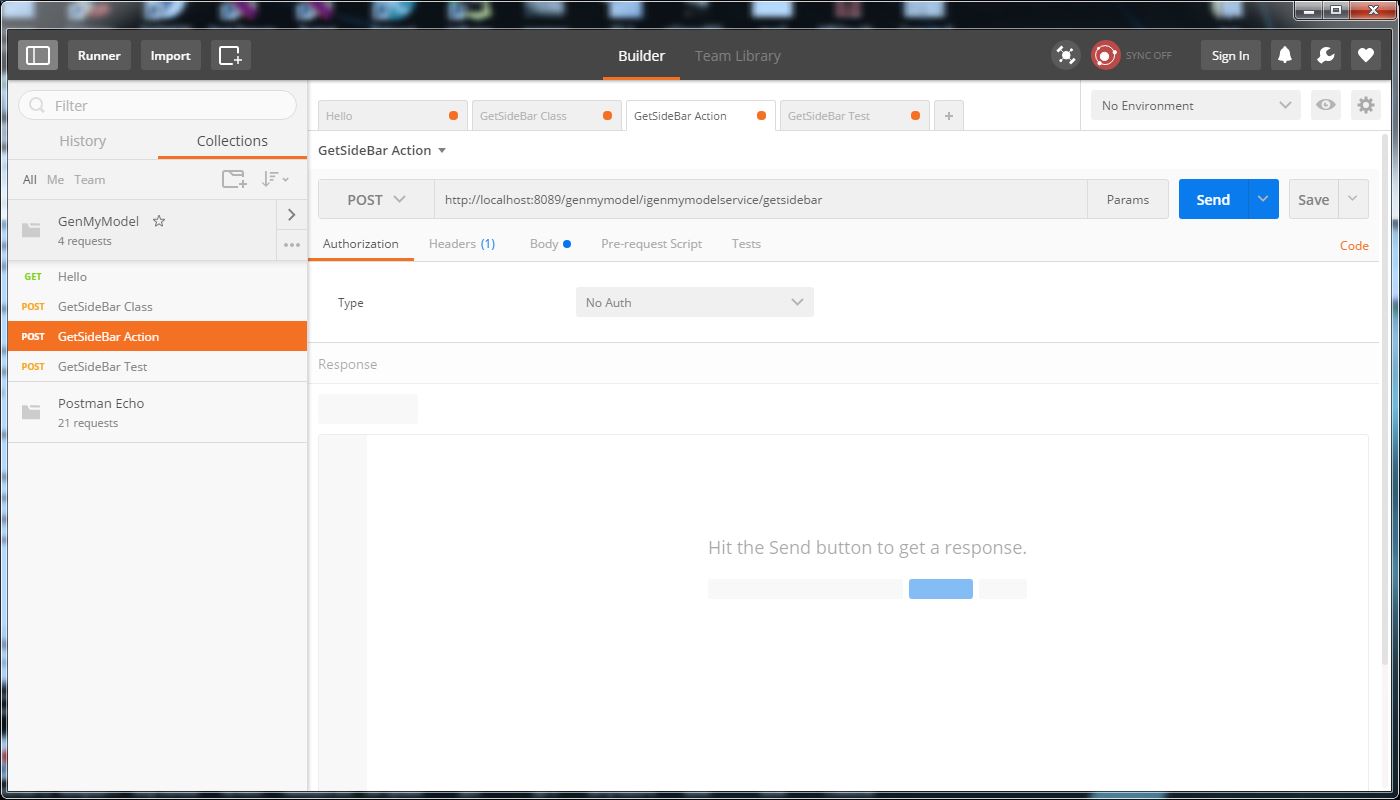


Рис 5. После вызова активной диаграммы

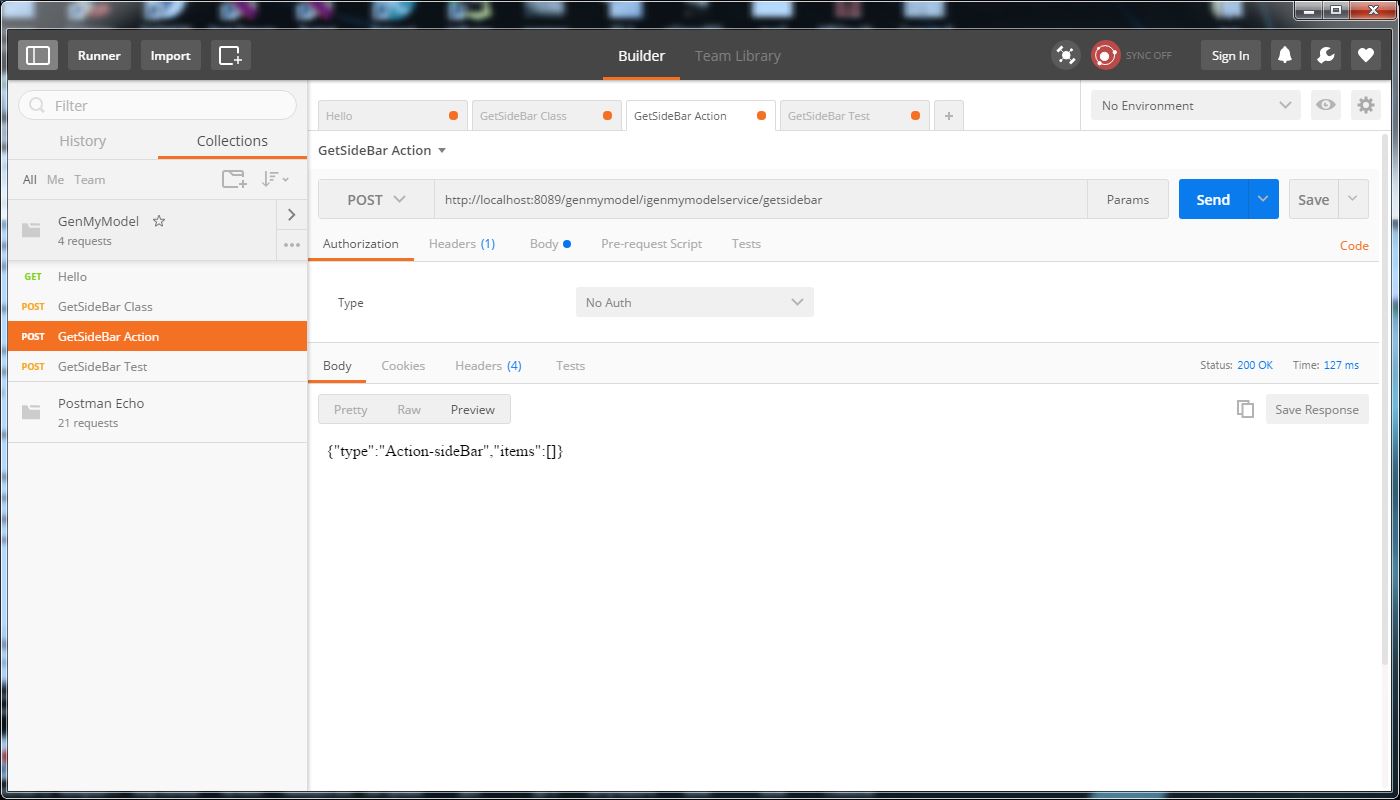
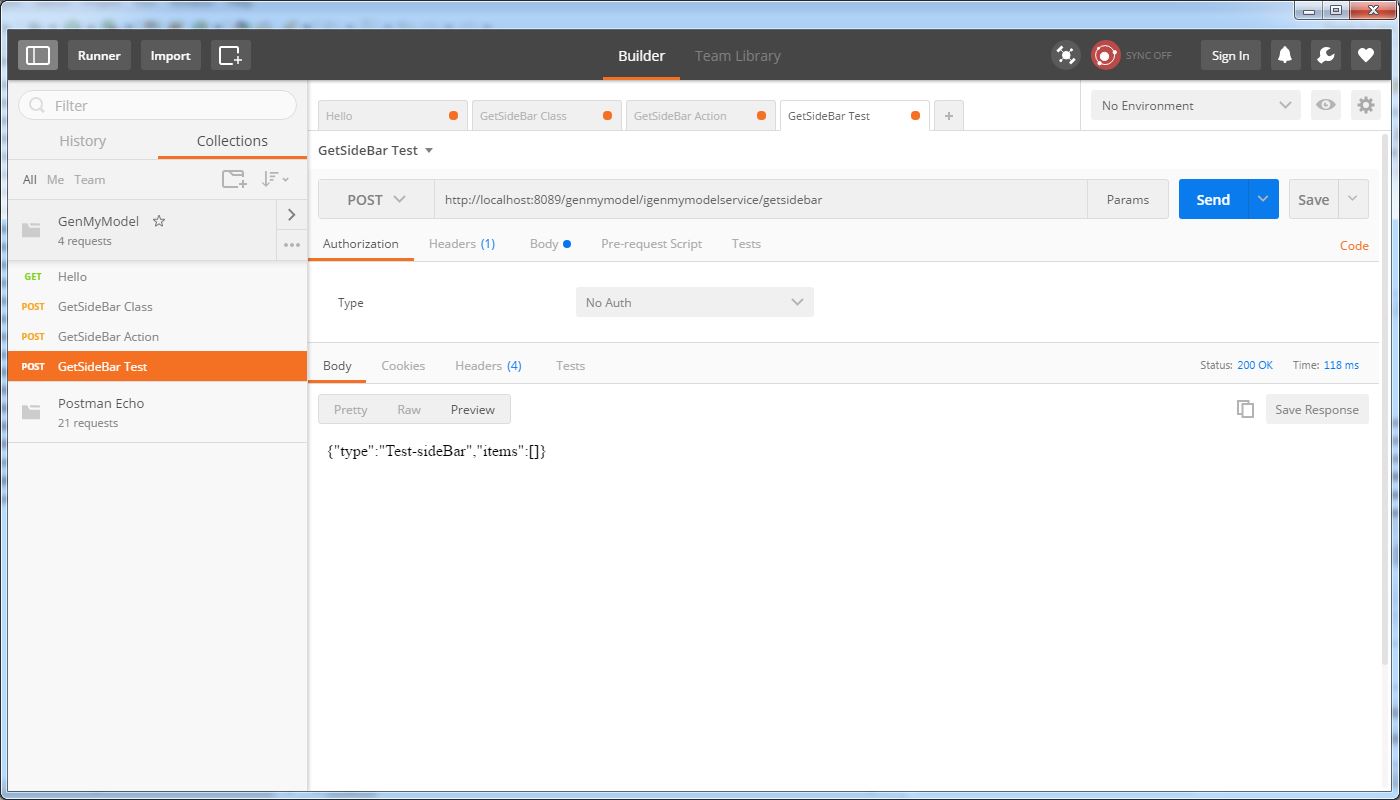


Рис 6. Тестовая диаграмма.



Приложение 1.

**package** ru.agentlab.genmymodel.ds.host;

**import** java.io.IOException;

**import** java.util.Dictionary;

**import** org.osgi.service.cm.ConfigurationException;

**import** org.osgi.service.cm.ManagedService;

**import** org.osgi.service.component.ComponentContext;

**import** org.osgi.service.component.annotations.Activate;

**import** org.osgi.service.component.annotations.Component;

**import** org.osgi.service.component.annotations.Deactivate;

**import** org.osgi.service.component.annotations.Modified;

**import** ru.agentlab.genmymodel.IGenMyModelService;

**import** ru.agentlab.genmymodel.model.Diagram;

**import** ru.agentlab.genmymodel.model.SideBar;

/\*\*

\*

\* Oomph component realisation.

\*

\* **@author** Sidorenko

\*

\*/

@Component(enabled = **true**, immediate = **true**,

property = { "service.exported.interfaces=\*", "service.exported.configs=ecf.jaxrs.jersey.server",

"ecf.jaxrs.jersey.server.urlContext=http://localhost:8080", "ecf.jaxrs.jersey.server.alias=/genmymodel",

"service.pid=ru.agentlab.genmymodel.ds.host.GenMyModelComponent" })

**public** **class** GenMyModelComponent

**implements** IGenMyModelService, ManagedService {

@Override

**public** String hello() {

**return** "Hello from GenMyModel";

}

@Override

**public** SideBar getSideBar(Diagram diagram) {

**switch** (diagram.getType()) {

**case** Diagram.***CLASS\_DIAGRAM\_TYPE***:

**return** **new** SideBar(SideBar.***CLASS\_SIDE\_BAR\_TYPE***);

**case** Diagram.***ACTION\_DIAGRAM\_TYPE***:

**return** **new** SideBar(SideBar.***ACTION\_SIDE\_BAR\_TYPE***);

**case** Diagram.***TEST\_DIAGRAM\_TYPE***:

**return** **new** SideBar(SideBar.***TEST\_SIDE\_BAR\_TYPE***);

**default**:

**return** **new** SideBar(SideBar.***UNDEFINED\_SIDE\_BAR\_TYPE***);

}

}

@Activate

**public** **void** activate(ComponentContext context) **throws** IOException {

System.***out***.println("GenMyModel service started"); //$NON-NLS-1$

}

@Deactivate

**public** **void** deactivate(ComponentContext context) {

System.***out***.println("GenMyModel service stopped"); //$NON-NLS-1$

}

@Modified

**public** **void** modify() {

System.***out***.println("GenMyModel service modified"); //$NON-NLS-1$

}

@Override

**public** **void** updated(Dictionary<String, ?> properties) **throws** ConfigurationException {

// Does nothing

}

}

Приложение 2.

**package** ru.agentlab.genmymodel.model;

**import** java.util.ArrayList;

**import** java.util.Collection;

/\*\*

\*

\* Instance of this class represents side bar.

\*

\* **@author** Sidorenko

\*

\*/

**public** **class** SideBar {

/\*\*

\* Class sideBar type.

\*/

**public** **static** **final** String ***UNDEFINED\_SIDE\_BAR\_TYPE*** = "Undefined-sideBar";

/\*\*

\* Class sideBar type.

\*/

**public** **static** **final** String ***CLASS\_SIDE\_BAR\_TYPE*** = "Class-sideBar";

/\*\*

\* Action sideBar type.

\*/

**public** **static** **final** String ***ACTION\_SIDE\_BAR\_TYPE*** = "Action-sideBar";

/\*\*

\* Test sideBar type.

\*/

**public** **static** **final** String ***TEST\_SIDE\_BAR\_TYPE*** = "Test-sideBar";

**private** String type;

**private** Collection<Object> items;

/\*\*

\* Default constructor

\*/

**public** SideBar(String type) {

**this**.type = type;

items = **new** ArrayList<>();

}

/\*\*

\* **@return** the type

\*/

**public** String getType() {

**return** type;

}

/\*\*

\* **@return** the items

\*/

**public** Collection<Object> getItems() {

**return** items;

}

}

Приложение 3.

**package** ru.agentlab.genmymodel.model;

**import** java.util.ArrayList;

**import** java.util.Collection;

/\*\*

\*

\* Instance of this class represents diagram.

\*

\* **@author** Sidorenko

\*

\*/

**public** **class** Diagram {

/\*\*

\* Class diagram type.

\*/

**public** **static** **final** String ***CLASS\_DIAGRAM\_TYPE*** = "Class-diagram";

/\*\*

\* Action diagram type.

\*/

**public** **static** **final** String ***ACTION\_DIAGRAM\_TYPE*** = "Action-diagram";

/\*\*

\* Test diagram type.

\*/

**public** **static** **final** String ***TEST\_DIAGRAM\_TYPE*** = "Test-diagram";

**private** **static** **final** String ***DEFAULT\_NAME*** = "Diagram";

**private** String name;

**private** String type;

**private** Collection<Object> diagramObjects;

/\*\*

\* Default constructor

\*/

**public** Diagram() {

name = ***DEFAULT\_NAME***;

type = ***CLASS\_DIAGRAM\_TYPE***;

diagramObjects = **new** ArrayList<>();

}

/\*\*

\* **@return** the name

\*/

**public** String getName() {

**return** name;

}

/\*\*

\* **@return** the type

\*/

**public** String getType() {

**return** type;

}

/\*\*

\* **@return** the diagramObjects

\*/

**public** Collection<Object> getDiagramObjects() {

**return** diagramObjects;

}

/\*\*

\* **@param** name the name to set

\*/

**public** **void** setName(String name) {

**this**.name = name;

}

/\*\*

\* **@param** type the type to set

\*/

**public** **void** setType(String type) {

**this**.type = type;

}

/\*\*

\* **@param** diagramObjects the diagramObjects to set

\*/

**public** **void** setDiagramObjects(Collection<Object> diagramObjects) {

**this**.diagramObjects = diagramObjects;

}

}

Приложение 4.

**package** ru.agentlab.genmymodel;

**import** ru.agentlab.genmymodel.model.Diagram;

**import** ru.agentlab.genmymodel.model.SideBar;

/\*\*

\*

\* GenMyModel service. Returns different type of side bars depending on diagram type.

\*

\* **@author** Sidorenko

\*

\*/

**public** **interface** IGenMyModelService {

/\*\*

\*

\* Says hello to user.

\*

\* **@return** hello message. Can't return <code>null</code>.

\*/

String hello();

/\*\*

\*

\* Returns side bar. Type of side bar depends on given diagram type.

\*

\* **@param** diagram - diagram. Can't be <code>null</code>.

\*

\* **@return** side bar in html format. Can't return <code>null</code>.

\*/

SideBar getSideBar(Diagram diagram);

}