

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет информационных

технологий, механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа № 5

По дисциплине «Инструментальные средства разработки ПО»

Выполнил студент группы М32011

Галстян Леон Ервандович

Проверил:

Приискалов Роман Андреевич

САНКТ-ПЕТЕРБУРГ

2021

О плагинах в IntelliJ Platform

Плагины – это единственный поддерживаемый путь расширения функциональности IDE на IntelliJ Platform. Плагин использует предоставляемый программный интерфейс среды или других плагинов для реализации собственных функциональных возможностей. Создать плагин можно в IntelliJ IDEA.

Совместимость с разными IDE

Все продукты, базирующиеся на IntelliJ Platform (IntelliJ IDEA, RubyMine, WebStorm, PhpStorm, PyCharm и AppCode) разделяют общий API платформы. Таким образом, плагины, не использующие какую-либо специфичную Java функциональность могут быть помечены как совместимые с другими продуктами, а не только с самой IDEA. Сделать это можно, определив зависимости от модулей в файле plugin.xml:

Если мы хотим, чтобы плагин работал на всех IntelliJ продуктах, определяем зависимости от модулей, доступных во всех продуктах: `com.intellij.modules.platform`, `com.intellij.modules.lang`, `com.intellij.modules.vcs`, `com.intellij.modules.xml`, `com.intellij.modules.xdebugger` (при этом хоть одну нужно определить). Пример: в создаваемом плагине в файле plugin.xml сразу есть зависимость:

```
22  on how to target different products -->
23  <depends>com.intellij.modules.platform</depends>
24
```

Если хотим создать плагин для конкретной IDE – нужно определить модули:

- `com.intellij.modules.java` – IntelliJ IDEA;
- `com.intellij.modules.ruby` – RubyMine;
- `com.intellij.modules.python` – PyCharm;
- `com.intellij.modules.objc` – AppCode.

Если плагин работает со всеми продуктами, но предоставляет некоторую специфичную функциональность для какого-то языка (например, java), то можно использовать следующий тег:

```
<depends optional="true"
```

```
config-file="my-java-features.xml">com.intellij.modules.java</depends>
```

Структура плагина

Помимо классов (на Java или Kotlin), реализующих функциональность, плагин должен иметь конфигурационный файл plugin.xml, который определяет имя плагина, описание, данные о разработчике, поддерживаемую версию IDE, компоненты, действия, группы действий и т.д.

IntelliJ IDEA предоставляет концепт действий (actions).

Действие – это класс, наследуемый от `AnAction`, чей метод `actionPerformed()` вызывается, когда выбран элемент меню или кнопка тулбара (реагирует на определенный триггер)

Система действий позволяет плагинам добавлять собственные пункты в меню и тулбары IDEA. Действие – это класс, унаследованный от `AnAction`, чей метод `actionPerformed()` вызывается, когда выбран элемент меню или соответствующая ему кнопка тулбара. Например, один и тот же класс действия отвечает за пункт меню «File | Open File...» и кнопку «Open File» на тулбаре.

Действия можно объединять в группы, которые, в свою очередь, могут содержать подгруппы (принадлежность действия к группе определяет, в каком месте интерфейса будет находиться действие.

Каждому действию или группе должен быть присвоен уникальный идентификатор. Идентификаторы большинства стандартных действий определены в классе `com.intellij.openapi.actionSystem.IdeActions`.

Каждое действие может быть включено в несколько групп, таким образом, они способны отображаться в различных местах пользовательского интерфейса. Места, в которых могут отображаться действия, определены как константы в интерфейсе `ActionPlaces` (из того же пакета). Для каждого места создается экземпляр класса `Presentation`, поэтому одно и то же действие может иметь различный текст или иконку в разных местах интерфейса.

Плагин OpenForums

[Ссылка на репозиторий](#)

Идея плагина

Очень часто при написании кода возникают проблемы или вопросы, ответы на которые ищутся в интернете. Чаще всего ответы находятся на форумах (и чаще всего, это несколько самых популярных форумов). Поэтому было бы неплохо сделать плагин, позволяющий облегчить процесс поиска.

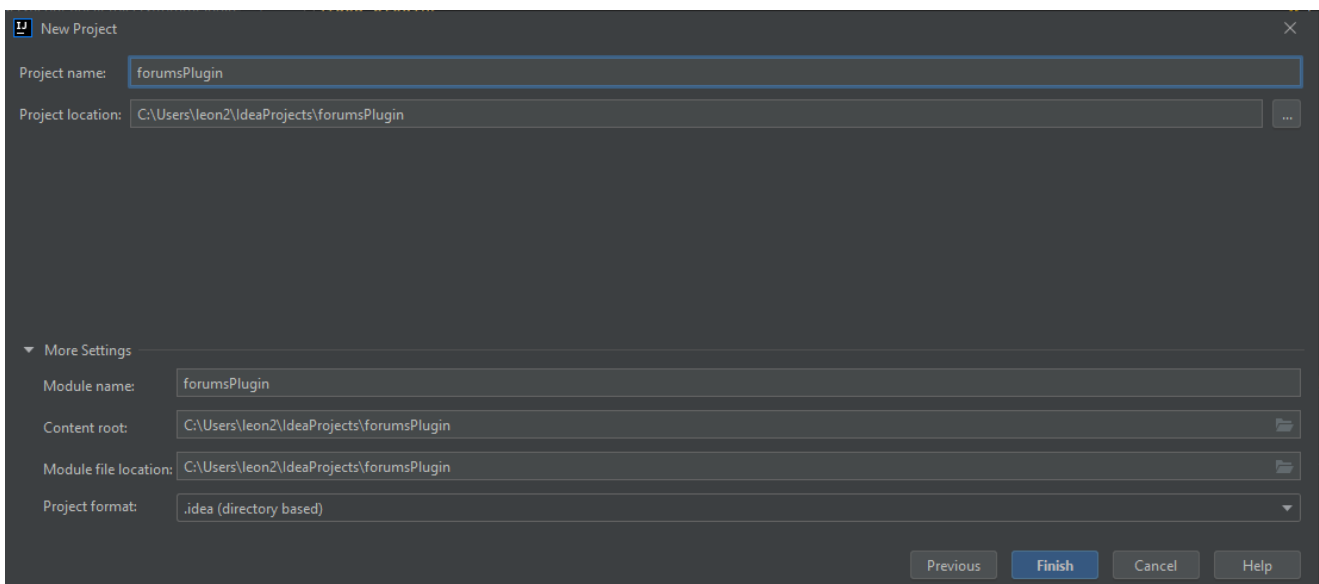
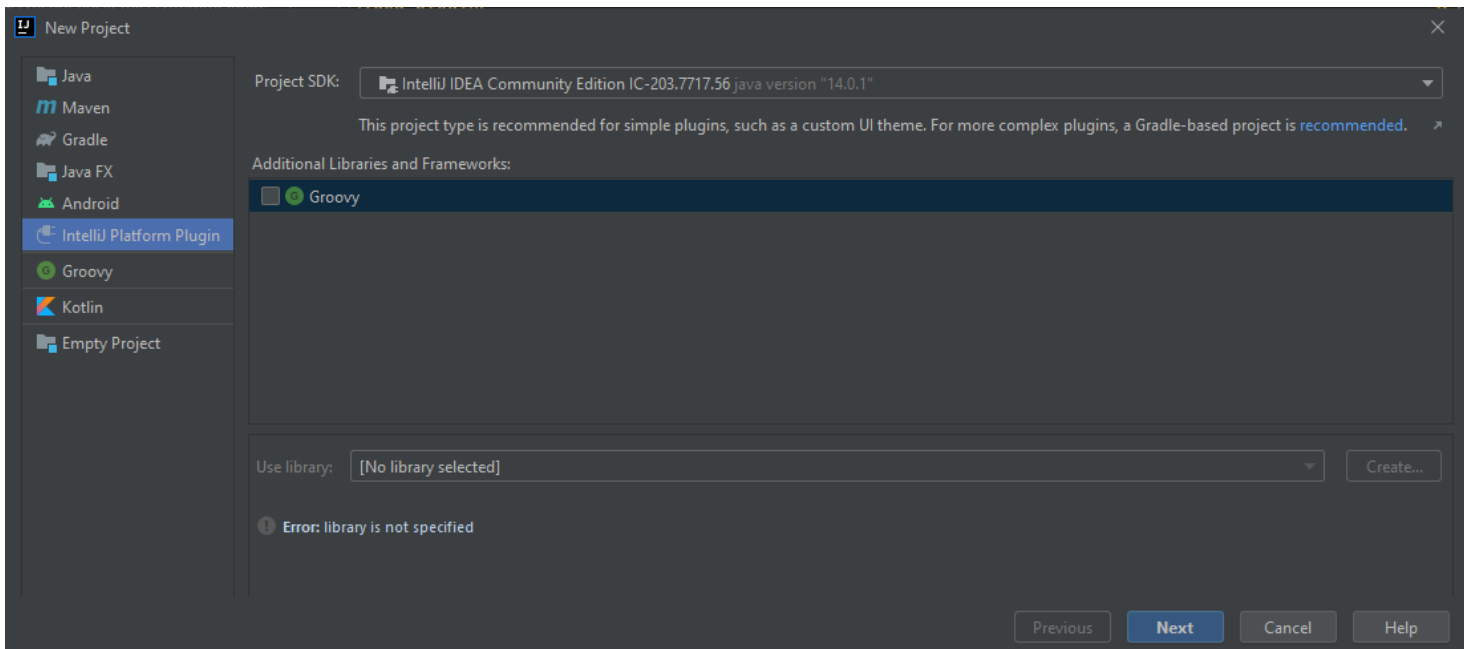
В плагине при нажатии комбинации клавиш будет открываться окошко, в котором вводится текст, а далее в браузере открываются вкладки форумов с уже произведенным поиском по этому тексту.

Создание плагина

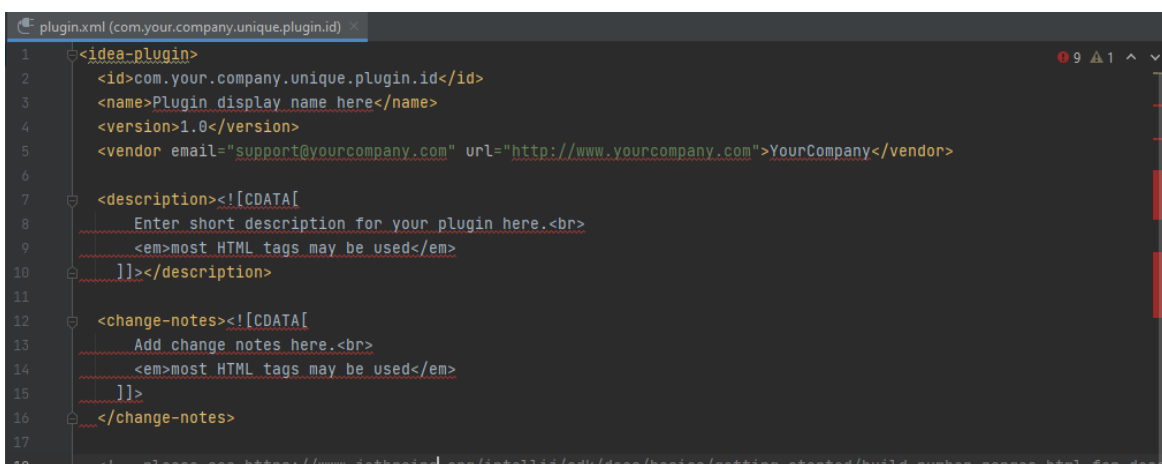
Для создания плагина требуется только IntelliJ IDEA (ну и плюс Java или Kotlin, я использовал Java)

Нужно создать проект плагина: New Project -> IntelliJ Platform Plugin

В качестве Project SDK можно выбрать любой продукт, базирующийся на IntelliJ Platform.



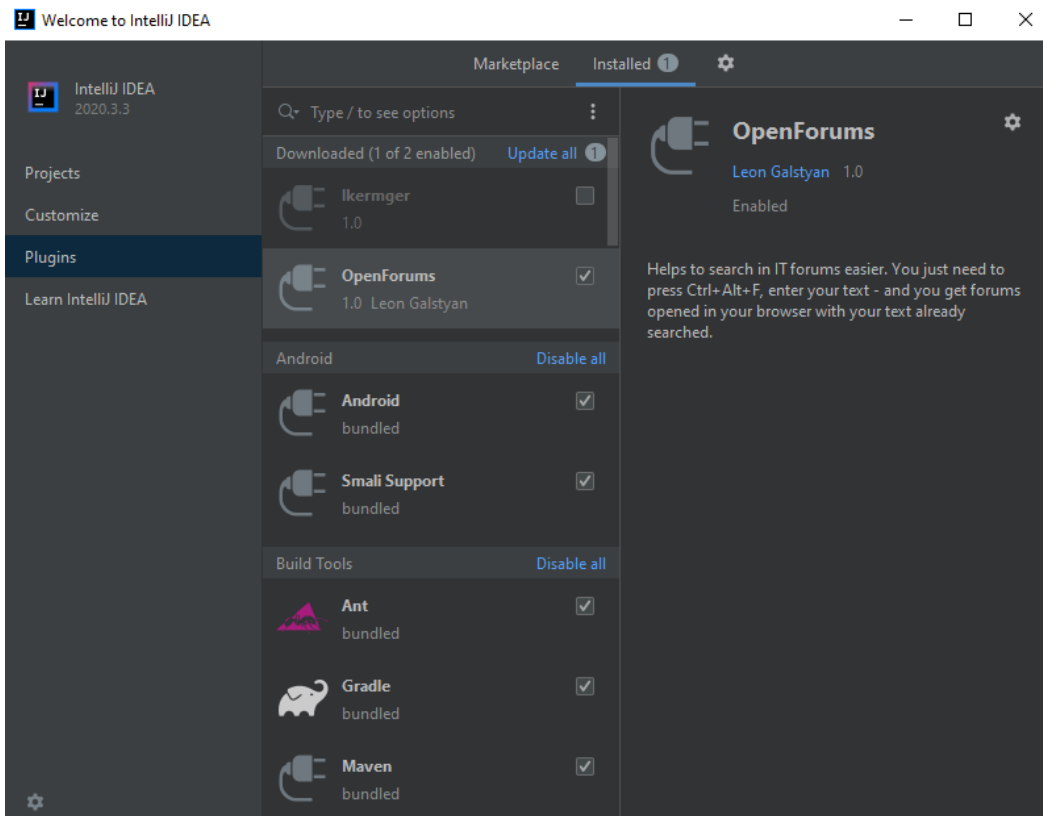
Создается и открывается файл plugin.xml. Нужно заполнить его поля – информацию о себе.



```
plugin.xml (com.leon.forumsPlugin)
1 <idea-plugin>
2   <id>com.leon.forumsPlugin</id>
3   <name>OpenForums</name>
4   <version>1.0</version>
5   <vendor email="leon25022002@gmail.com" url="https://github.com/Li-0-H">Leon Galstyan</vendor>
6
7   <description><![CDATA[
8     Helps to search in IT forums easier.
9     You just need to press Ctrl+Alt+F, enter your text -
10    and you get forums opened in your browser with your text already searched.
11  ]]></description>
12
13  <change-notes><![CDATA[
14
15  ]]>
16 </change-notes>
17
```

Заметим, что у нас изначально есть готовая конфигурация запуска проекта.

Пробуем запустить плагин. Открывается IntelliJ IDEA (ее мы указали в качестве SDK), в которой установлен плагин:



Теперь реализуем желаемый функционал. Создадим Action:

Заметим, что в plugin.xml создается запись о созданном Action:

```
29 <actions>
30   <!-- Add your actions here -->
31   <action id="OpenForums" class="OpenForumsAction" text="Open forums"
32     description="Contains the whole logic of plugin">
33     <keyboard-shortcut keymap="$default" first-keystroke="ctrl alt F"/>
34   </action>
35 </actions>
36
```

Заполним класс функционалом

```
import com.intellij.ide.BrowserUtil;
import com.intellij.openapi.actionSystem.AnAction;
import com.intellij.openapi.actionSystem.AnActionEvent;
import com.intellij.openapi.ui.Messages;

import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;
import java.util.List;

public class OpenForumsAction extends AnAction {

    public enum Forums {

        STACK_OVERFLOW {
            @Override
            public String getUrl(String textToSearch) {
                String encodedText = URLEncoder.encode(textToSearch,
StandardCharsets.UTF_8);
                return String.format("https://stackoverflow.com/search?q=%s",
encodedText);
            }
        },

        HABR_QNA {
            @Override
            public String getUrl(String textToSearch) {
                String encodedText = URLEncoder.encode(textToSearch,
StandardCharsets.UTF_8);
                return String.format("https://qna.habr.com/search?q=%s", encodedText);
            }
        },

        GOOGLE {
            @Override
            public String getUrl(String textToSearch) {
                String encodedText = URLEncoder.encode(textToSearch,
StandardCharsets.UTF_8);
                return String.format("https://www.google.com/search?q=%s",
encodedText);
            }
        },

        STACK_OVERFLOW_RU{
            @Override
            public String getUrl(String textToSearch) {
                String encodedText = URLEncoder.encode(textToSearch,
StandardCharsets.UTF_8);
                return String.format("https://ru.stackoverflow.com/search?q=%s",
encodedText);
            }
        },
    }
}
```

```

        CYBERFORUM{
            @Override
            public String getUrl(String textToSearch) {
                String encodedText = URLEncoder.encode(textToSearch,
StandardCharsets.UTF_8);
                return String.format("https://www.cyberforum.ru/google.php?q=%s",
encodedText);
            }
        },

        CODEPROJECT{
            @Override
            public String getUrl(String textToSearch) {
                String encodedText = URLEncoder.encode(textToSearch,
StandardCharsets.UTF_8);
                return String.format("https://www.codeproject.com/search.aspx?q=%s",
encodedText);
            }
        };

        public abstract String getUrl(String textToSearch);
    }

    public List<Forums> forumsList = List.of
        (Forums.GOOGLE, Forums.STACK_OVERFLOW, Forums.HABR_QNA, Forums.CYBERFORUM,
Forums.STACK_OVERFLOW_RU, Forums.CODEPROJECT);

    @Override
    public void actionPerformed(ActionEvent e) {
        String textToSearch = Messages.showInputDialog("Enter the text you want to
search",
            "OpenForums", Messages.getQuestionIcon());
        if (textToSearch == null) {
            return;
        }
        if (textToSearch.isEmpty()) {
            Messages.showMessageDialog("Error: Text to search can't be empty",
                "OpenForums", Messages.getErrorIcon());
            return;
        }
        for (Forums forum : forumsList) {
            BrowserUtil.browse(forum.getUrl(textToSearch));
        }
    }
}

```

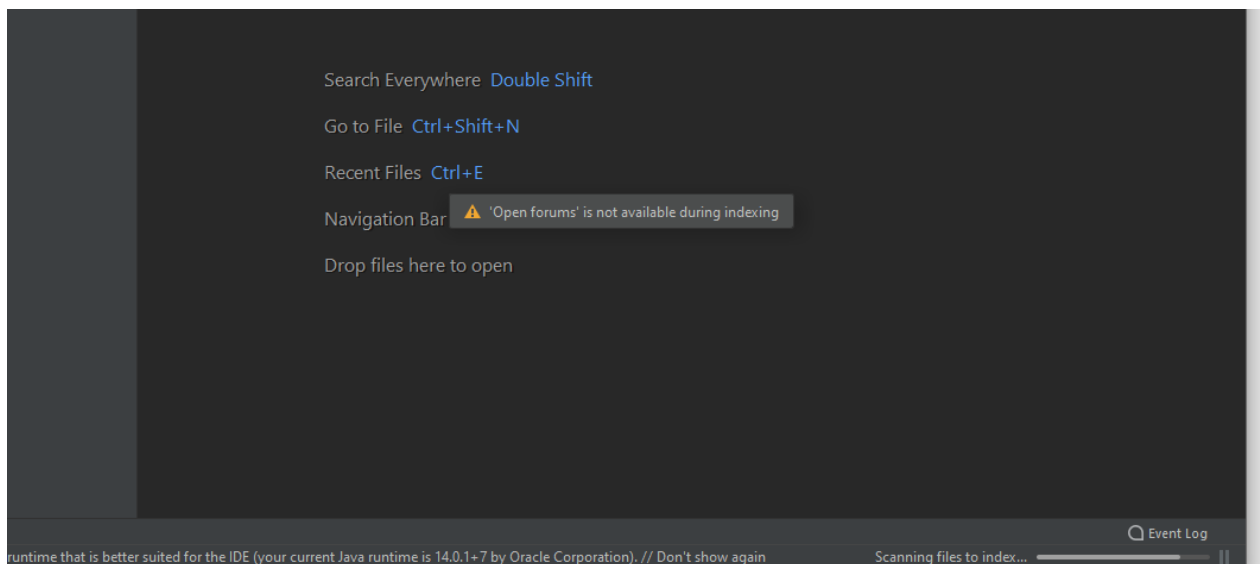
BrowserUtil.browse(url) – в браузере по умолчанию выполняется запрос по адресу url

Messages.showInputDialog – показывает диалог с полем для текста, возвращает введенную строку (null – если нажали cancel)

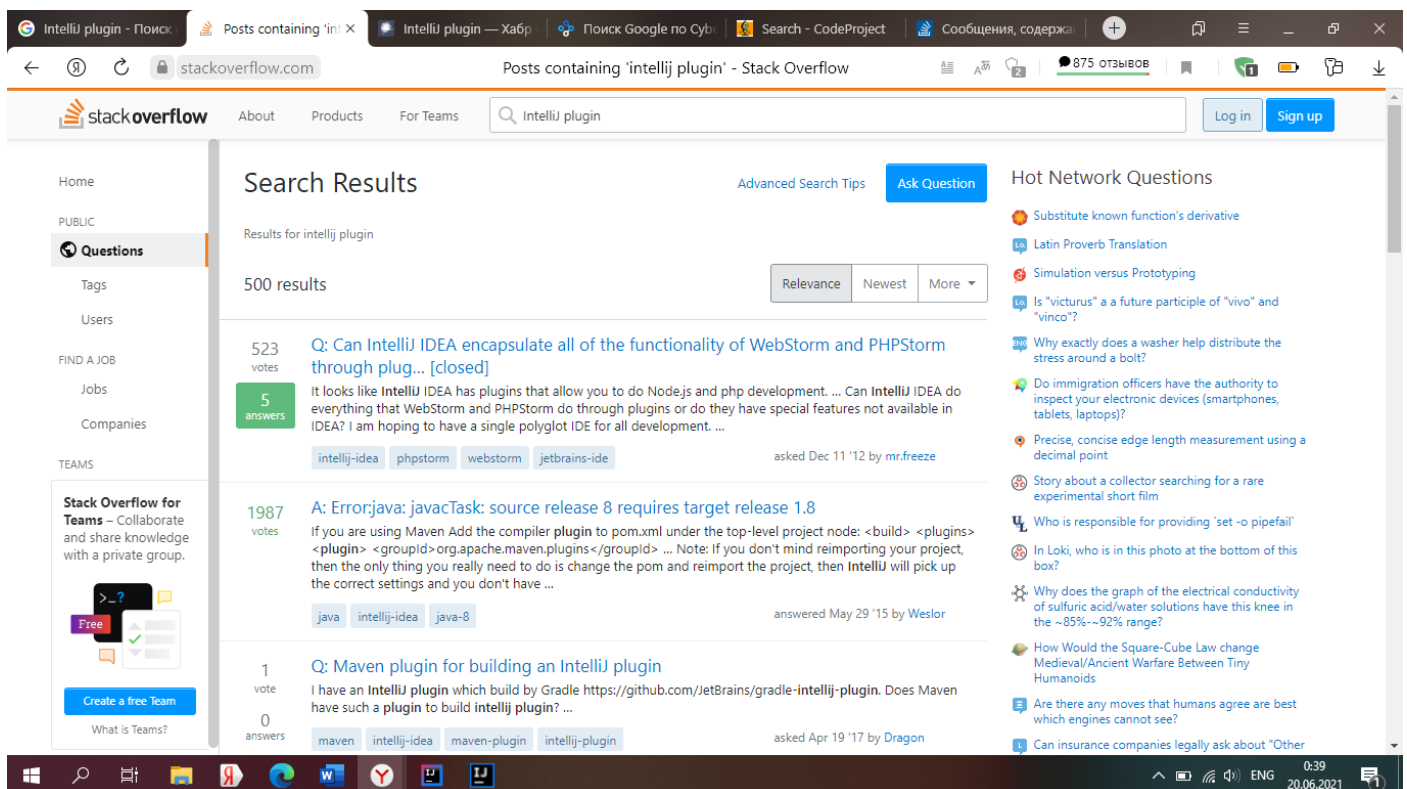
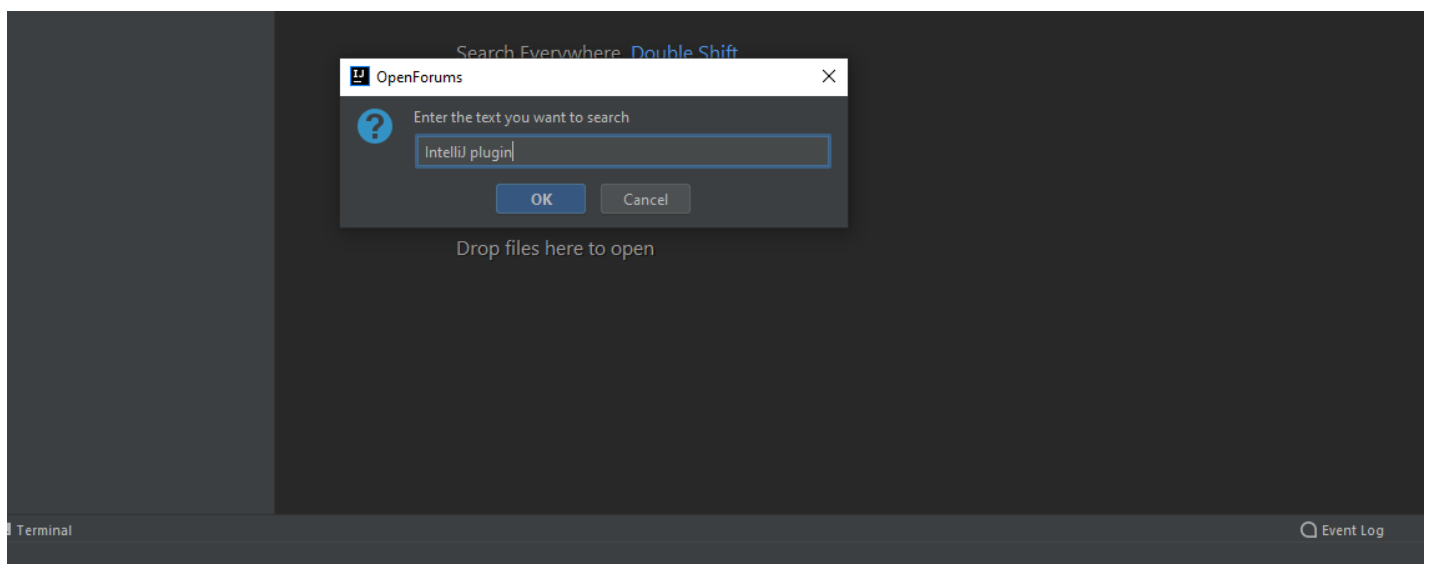
Messages.showMessageDialog – простое текстовое сообщение

URLEncoder.encode – кодирует строку так, чтобы она соответствовала формату url

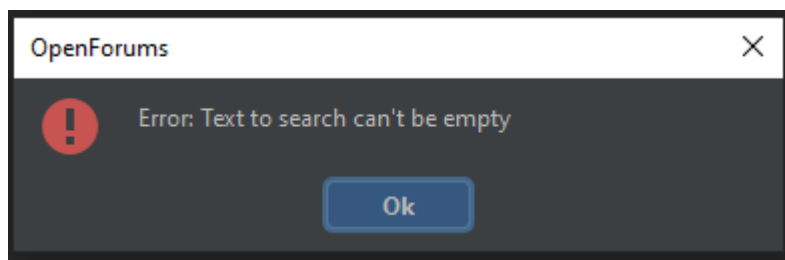
Попробуем запустить и проверить функциональность



Все работает правильно, кроме того, что плагин не работает во время индексирования.



При вводе пустой строки:



Чтобы плагин работал и во время индексирования, нужно добавить в Action реализацию метода `isDumbAware`

```
@Override
public boolean isDumbAware() {
    return true;
}
```

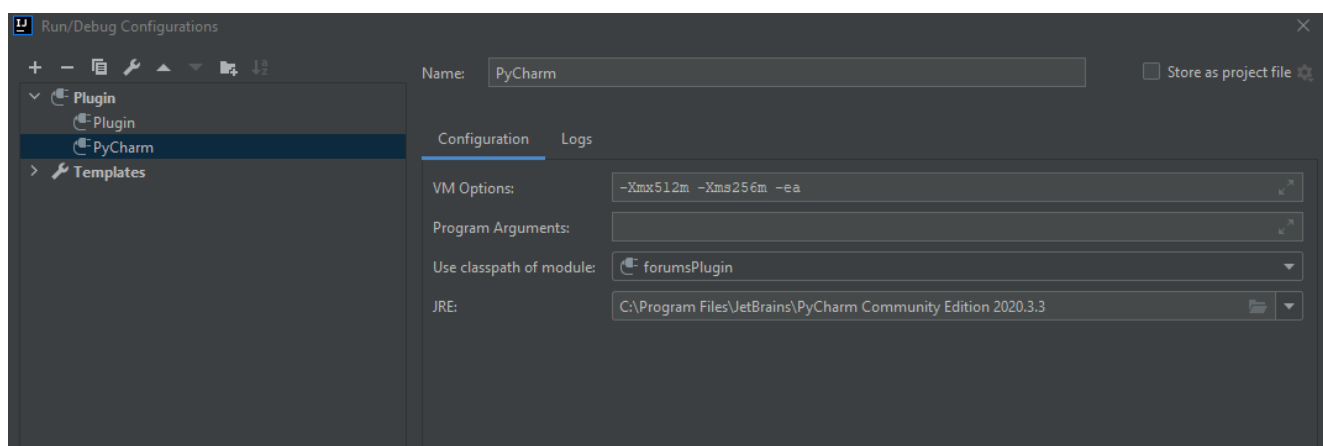
Также мне не понравилось, что пользователь может менять размер диалоговых окон, и я это исправил

Теперь создание диалогового окна выглядит так:

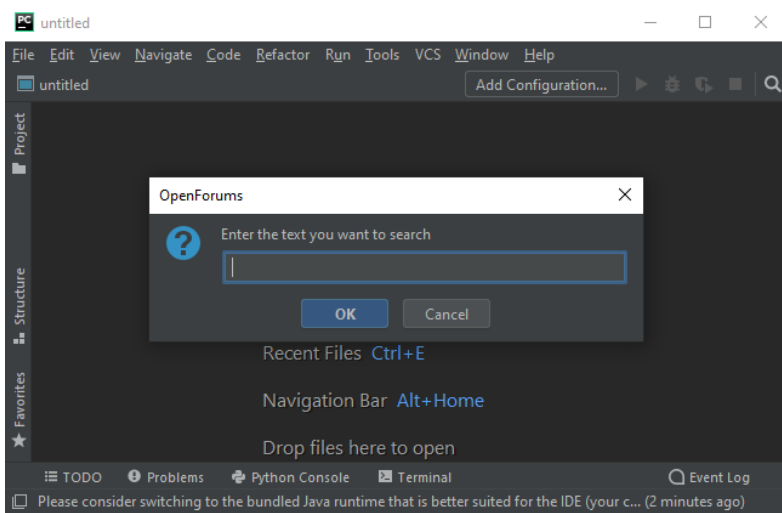
```
15 of public void actionPerformed(ActionEvent e) {
16     Messages.InputDialog textInput = new Messages.InputDialog( message: "Enter the text you want to search",
17         title: "OpenForums", Messages.getQuestionIcon(), initialValue: null, validator: null);
18     textInput.setResizable(false);
19     textInput.setSize( width: 1000, height: 350);
20     textInput.show();
21     String textToSearch = textInput.getInputString();
22     if (textToSearch == null) {
```

Размер окна задан изначально и задан неизменяемым.

Проверим, что плагин работает не только для IDEA. Создадим новую конфигурацию:



и запустим:



Следующее, что я решил поменять – вынести объявление перечисления Forums в отдельный файл

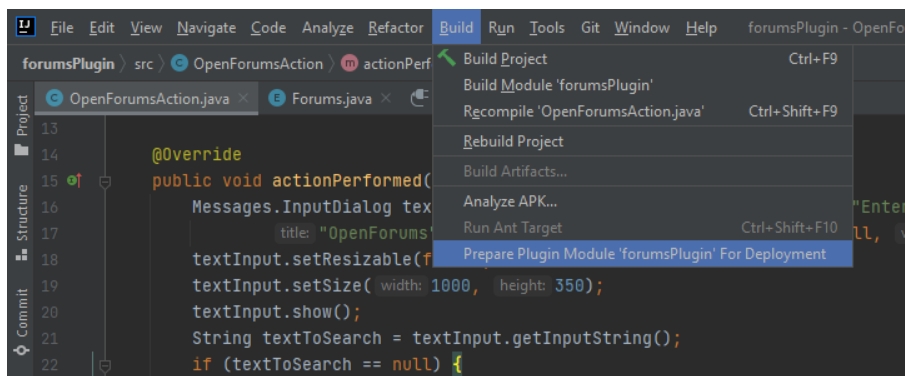
Далее я добавил историю поиска – можно кнопками вверх-вниз, как консоли, переключаться по истории (пока история хранится только при работающей IDE)

```
textInput.getTextField().addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
    }

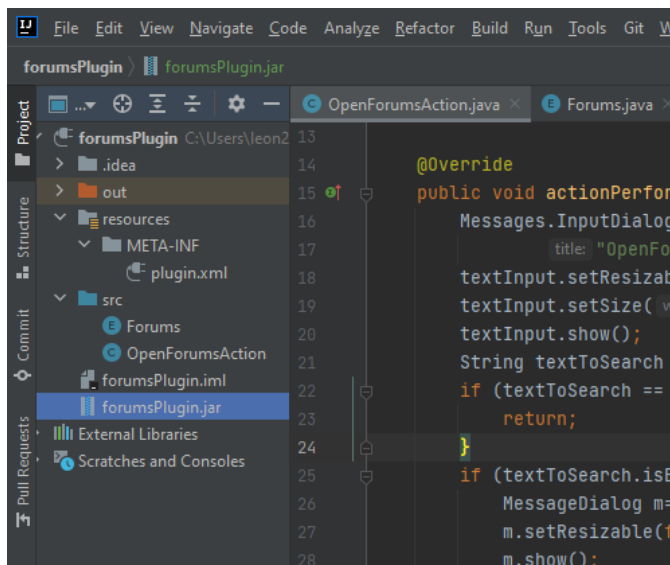
    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_UP) {
            if (nowOnIndex == -1) {
                beforePressingUp[0] = textInput.getTextField().getText();
            }
            if (!latestSearches.isEmpty() && nowOnIndex + 1 <= latestSearches.size() -
1) {
                nowOnIndex++;
                textInput.getTextField().setText(latestSearches.get(nowOnIndex));
            }
        }
        if (e.getKeyCode() == KeyEvent.VK_DOWN) {
            if (nowOnIndex == 0) {
                nowOnIndex--;
                textInput.getTextField().setText(beforePressingUp[0]);
            }
            if (!latestSearches.isEmpty() && nowOnIndex - 1 >= 0) {
                nowOnIndex--;
                textInput.getTextField().setText(latestSearches.get(nowOnIndex));
            }
        }
    }

    @Override
    public void keyReleased(KeyEvent e) {
    }
});
```

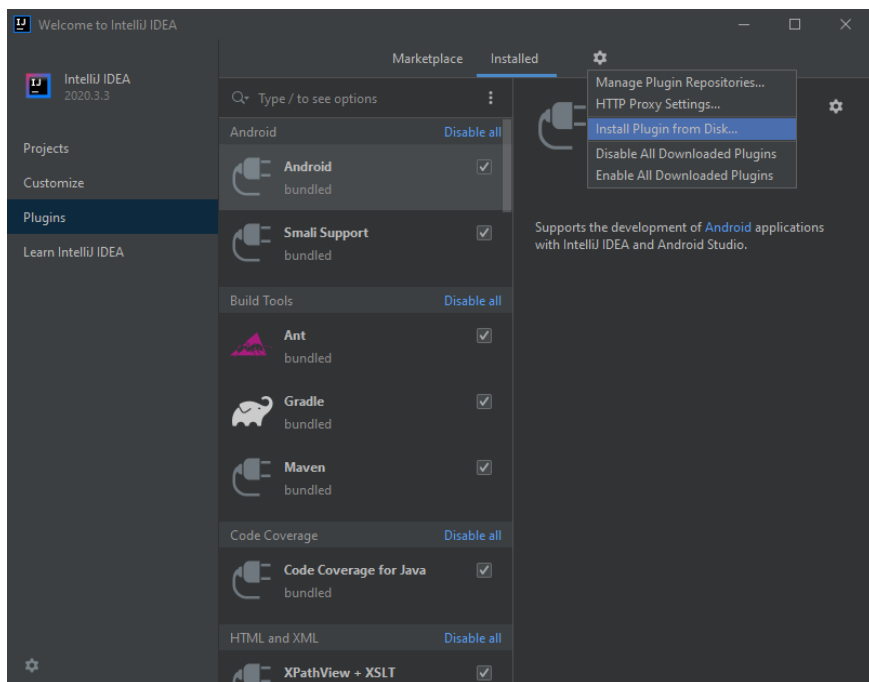
Далее, чтобы получить готовый плагин, нужно собрать его в .jar файл



В папке проекта появился файл:

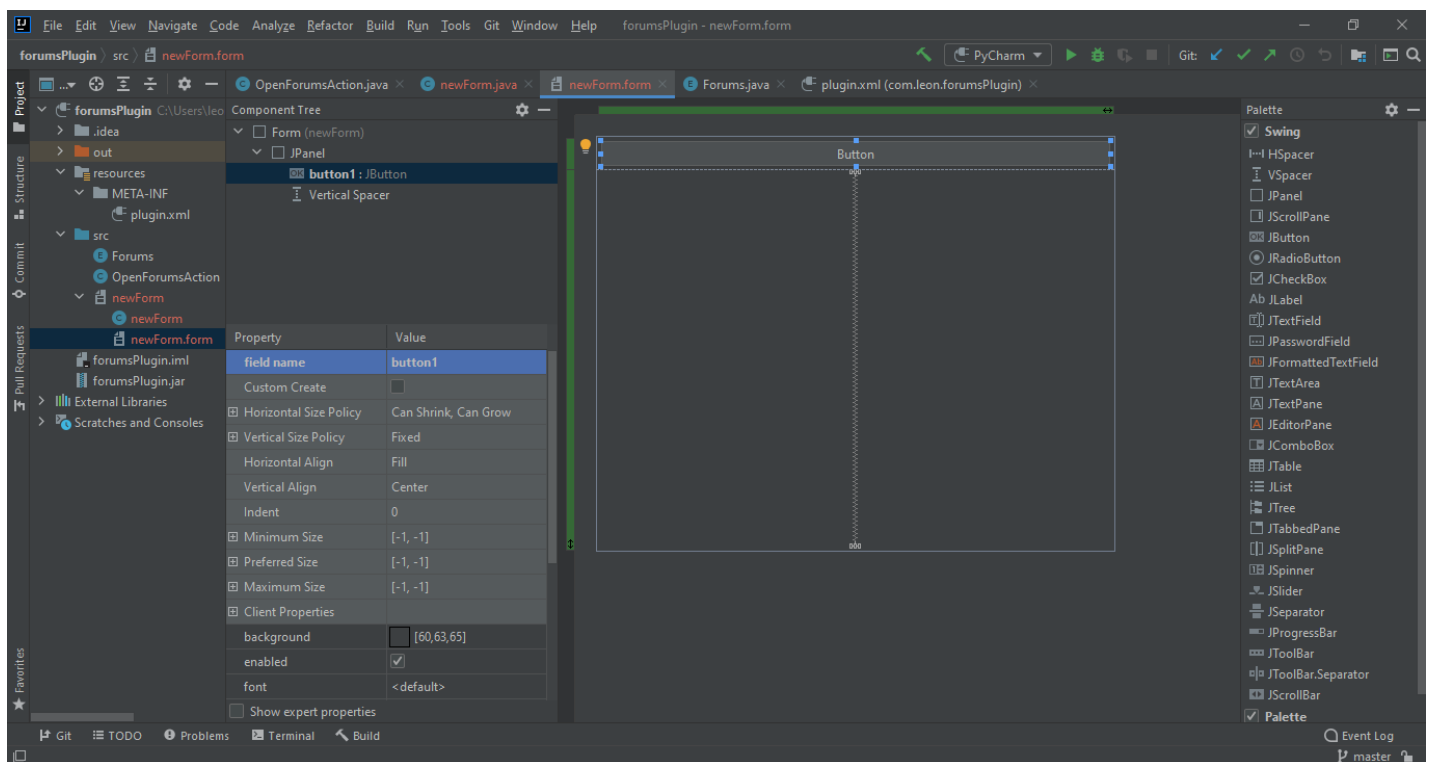
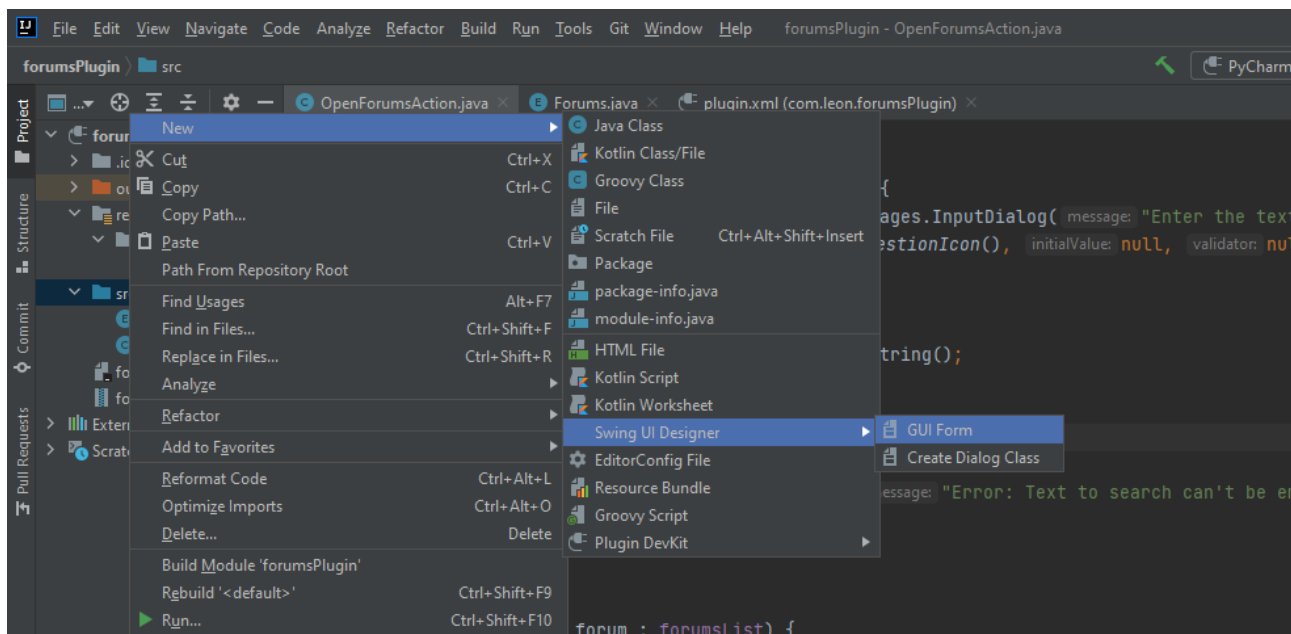


Далее заходим в IDE, в которой мы хотим использовать плагин, и устанавливаем его:



Что еще умеет IDEA

Очень удобный графический редактор для создания окон



Выводы

Написание (в том числе запуск и отладка) плагинов для платформы IntelliJ не требует каких-то особых навыков и установки особого ПО. IDEA предоставляет для этого удобный интерфейс с множеством графических интерфейсов и с богатыми библиотеками. Но, к сожалению, документации на все это крайне мало (точнее, она совсем не подробная), так что во многом приходится действовать наугад, ориентируясь на код имеющихся плагинов.